

JP9034723

Publication Title:

COMMITMENT OF INTRODUCTION PLAN OBJECT FOR INTRODUCTION OF APPLICATION PROGRAM AT NETWORK

Abstract:

Abstract of JP 9034723

(A) Translate this text PROBLEM TO BE SOLVED: To introduce an application on a network by inspecting the suitability of an introduction schedule object in advance and transforming that object into data structure usable for a network introduction engine. SOLUTION: A commit process is started by selecting any commit menu out of an introduction schedule menu or the like and in a step 405, the introduction schedule object prepares an error log object for storing the respective messages of conditions, warning and error detected during the commit process. Next, in a step 410, when it is necessary to properly perform introduction or constitution, any object or attribute is added to the schedule object. Next, in a step 415, it is inspected whether proper data exist in the schedule or not. At a judge block 420, an error counter is inspected and when counting is '0', processing is moved to the part after a step 425 so that the object can be transformed into the data structure usable for the network introduction engine of application introduction.

Courtesy of <http://v3.espacenet.com>

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/445			G 0 6 F 9/06	4 2 0 J
9/06	4 1 0			4 1 0 B

審査請求 未請求 請求項の数25 O L (全 46 頁)

(21)出願番号 特願平8-82910
 (22)出願日 平成8年(1996)4月4日
 (31)優先権主張番号 4 1 7 1 6 2
 (32)優先日 1995年4月5日
 (33)優先権主張国 米国 (US)

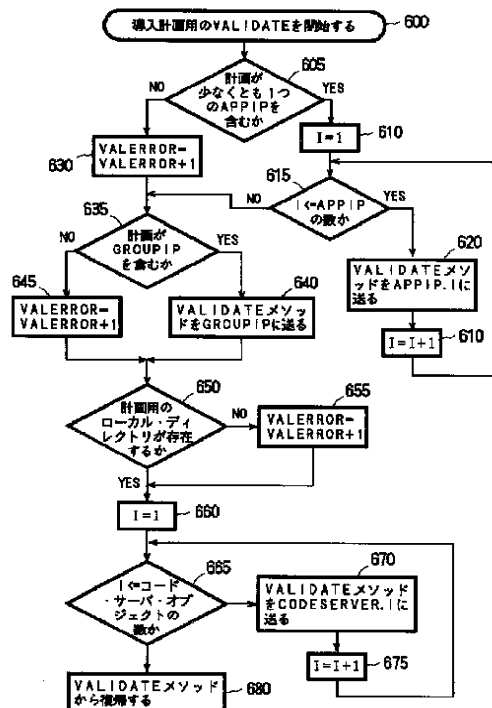
(71)出願人 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレーション
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 アメリカ合衆国10504、ニューヨーク州アーモンク (番地なし)
 (72)発明者 ジョン・ローレンス・バンス
 アメリカ合衆国78590 テキサス州オースチン バトンウッド 11500
 (74)代理人 弁理士 合田 潔 (外2名)

最終頁に続く

(54)【発明の名称】 ネットワークでのアプリケーション・プログラムの導入のための導入計画オブジェクトのコミット

(57)【要約】 (修正有)

【課題】 導入計画オブジェクトを妥当性検査する。
 【解決手段】 導入計画オブジェクトは、プログラムを表すアプリケーション・オブジェクトと、そのアプリケーション・プログラムを導入するワークステーションのグループを表すオブジェクトとを含む。コミット・プロセスの一部として、導入計画オブジェクトは、その子オブジェクトを検査し、必要であれば導入計画オブジェクトに子オブジェクトを追加することによって事前妥当性検査され、導入計画オブジェクトとその子オブジェクトのデータにエラーがないかどうか検査することによって妥当性検査され、ネットワーク内でアプリケーションを導入するネットワーク導入エンジンに使用可能なデータ構造に変換される。そのアプリケーションの導入が応答ファイルと、そのデータをカスタマイズするためのカスタマイズ・ファイル・オブジェクトとを必要とする場合、導入計画は応答ファイル・オブジェクトをさらに含む。



【特許請求の範囲】

【請求項1】ネットワークでアプリケーションを導入するための導入計画オブジェクトをコミットする方法において、

導入計画オブジェクト内の子オブジェクトを検査し、必要であれば、導入計画オブジェクトに追加の子オブジェクトを追加することにより、導入計画オブジェクトを事前妥当性検査するステップと、

導入計画オブジェクトとその子オブジェクト内のデータを検査して、データにエラーがないかどうか確認することにより、導入計画オブジェクトを妥当性検査するステップと、

導入計画が正常に妥当性検査された場合に、導入計画オブジェクト内の子オブジェクトを、データ構造に応じてネットワークでアプリケーションを導入するネットワーク導入エンジンに使用可能なデータ構造に変換するステップとを含む方法。

【請求項2】導入計画オブジェクトが、アプリケーション・プログラムを表す計画中アプリケーション・オブジェクトと、アプリケーション・プログラムを導入すべきワークステーションのグループを表す計画中グループ・オブジェクトとを含むことを特徴とする、請求項1に記載の方法。

【請求項3】導入計画が、アプリケーションの導入用のデータを含む応答ファイル・オブジェクトと、特定のワークステーション用の応答ファイル・オブジェクト・データをカスタマイズするためのデータを含むカスタマイズ・ファイル・オブジェクトとをさらに含むことを特徴とする、請求項2に記載の方法。

【請求項4】妥当性検査ステップが、導入計画オブジェクト内のデータとして指定されたターゲット・ワークステーション上のファイル・ディレクトリを検査して、そのファイル・ディレクトリが物理的に有効でネットワーク上でアクセス可能であるかどうかを判定することをさらに含み、ターゲット・ワークステーションが、そのアプリケーションを導入すべきワークステーションのグループのうちの1つであることを特徴とする、請求項2に記載の方法。

【請求項5】妥当性検査ステップが、ワークステーションのグループ上でのアプリケーションの導入に必要なコード・サーバ・ワークステーション上のオブジェクトの存在を確認することをさらに含み、コード・サーバ・オブジェクトが、導入計画オブジェクトの子オブジェクトではないことを特徴とする、請求項2に記載の方法。

【請求項6】妥当性検査ステップが、アプリケーション・オブジェクトが、ネットワーク内の第1のアプリケーションに更新内容を提供するアプリケーションを表していることを判定するステップと、アプリケーション・コンテナ・オブジェクト内の第1のアプリケーションを表す第1のアプリケーション・オブ

ジェクトの存在を確認するステップとをさらに含むことを特徴とする、請求項2に記載の方法。

【請求項7】妥当性検査ステップが、ターゲット・ワークステーション上のオペレーティング・システムとその訂正サービスのレベルとを確認することをさらに含むことを特徴とする、請求項2に記載の方法。

【請求項8】導入計画オブジェクトが、導入すべき複数のアプリケーションと対応する応答ファイルとをそれぞれ表す、複数の計画中アプリケーション・オブジェクトと複数の対応する応答ファイル・オブジェクトとを含み、変換ステップが、ワークステーションのグループ内の特定のワークステーションに応じて複数の応答ファイル・オブジェクトをカスタマイズするために、カスタマイズ・ファイル・オブジェクトを複数の応答ファイル・オブジェクトに適用することをさらに含むことを特徴とする、請求項3に記載の方法。

【請求項9】複数の導入計画オブジェクトについて、事前妥当性検査ステップと、妥当性検査ステップと、変換ステップとを繰り返すステップと、各ワークステーションごとにワークステーション待ち行列に複数の導入計画オブジェクトを待ち行列化するステップとをさらに含むことを特徴とする、請求項1に記載の方法。

【請求項10】ネットワークでアプリケーションを導入するための導入計画オブジェクトをコミットするシステムにおいて、

導入計画オブジェクト内の子オブジェクトを検査し、必要であれば、導入計画オブジェクトに追加の子オブジェクトを追加することにより、導入計画オブジェクトを事前妥当性検査する手段と、

導入計画オブジェクトとその子オブジェクト内のデータを検査して、データにエラーがないかどうか確認することにより、導入計画オブジェクトを妥当性検査する手段と、

導入計画が正常に妥当性検査された場合に、導入計画オブジェクト内の子オブジェクトを、データ構造に応じてネットワークでアプリケーションを導入するネットワーク導入エンジンに使用可能なデータ構造に変換する手段とを含むシステム。

【請求項11】導入計画オブジェクトが、アプリケーション・プログラムを表す計画中アプリケーション・オブジェクトと、アプリケーション・プログラムを導入すべきワークステーションのグループを表す計画中グループ・オブジェクトとを含むことを特徴とする、請求項10に記載のシステム。

【請求項12】導入計画が、アプリケーションの導入用のデータを含む応答ファイル・オブジェクトと、特定のワークステーション用の応答ファイル・オブジェクト・データをカスタマイズするためのデータを含むカスタマイズ・ファイル・オブジェクトとをさらに含むことを特

徴とする、請求項11に記載のシステム。

【請求項13】妥当性検査手段が、導入計画オブジェクト内のデータとして指定されたターゲット・ワークステーション上のファイル・ディレクトリを検査して、そのファイル・ディレクトリが物理的に有効でネットワーク上でアクセス可能であるかどうかを判定する手段をさらに含み、ターゲット・ワークステーションが、そのアプリケーションを導入すべきワークステーションのグループのうちの1つであることを特徴とする、請求項11に記載のシステム。

【請求項14】妥当性検査手段が、ワークステーションのグループ上でアプリケーションの導入に必要なコード・サーバ・ワークステーション上のオブジェクトの存在を確認する手段をさらに含み、コード・サーバ・オブジェクトが、導入計画オブジェクトの子オブジェクトではないことを特徴とする、請求項11に記載のシステム。

【請求項15】妥当性検査手段が、アプリケーション・オブジェクトが、ネットワーク内の第1のアプリケーションに更新内容を提供するアプリケーションを表していることを判定する手段と、アプリケーション・コンテナ・オブジェクト内の第1のアプリケーションを表す第1のアプリケーション・オブジェクトの存在を確認する手段とをさらに含むことを特徴とする、請求項11に記載のシステム。

【請求項16】導入計画オブジェクトが、導入すべき複数のアプリケーションと対応する応答ファイルとをそれぞれ表す、複数の計画中アプリケーション・オブジェクトと複数の対応する応答ファイル・オブジェクトとを含み、変換ステップが、ワークステーションのグループ内の特定のワークステーションに応じて複数の応答ファイル・オブジェクトをカスタマイズするために、カスタマイズ・ファイル・オブジェクトを複数の応答ファイル・オブジェクトに適用することをさらに含むことを特徴とする、請求項12に記載のシステム。

【請求項17】複数の導入計画オブジェクトについて、事前妥当性検査ステップと、妥当性検査ステップと、変換ステップとを繰り返す手段と、各ワークステーションごとにワークステーション待ち行列に複数の導入計画オブジェクトを待ち行列化する手段とをさらに含むことを特徴とする、請求項10に記載のシステム。

【請求項18】ネットワークでアプリケーションを導入するための導入計画オブジェクトをコミットするためにコンピュータで読取り可能な命令を格納するコンピュータ・メモリにおいて、導入計画オブジェクト内の子オブジェクトを検査し、必要であれば、導入計画オブジェクトに追加の子オブジェクトを追加することにより、導入計画オブジェクトを事前妥当性検査する手段と、

導入計画オブジェクトとその子オブジェクト内のデータを検査して、データにエラーがないかどうか確認することにより、導入計画オブジェクトを妥当性検査する手段と、

導入計画が正常に妥当性検査された場合に、導入計画オブジェクト内の子オブジェクトを、データ構造に応じてネットワークでアプリケーションを導入するネットワーク導入エンジンに使用可能なデータ構造に変換する手段とを含み、

コンピュータ・メモリがコンピュータ・システムに結合され、コンピュータ・システムによってアクセスされたときに、その手段が活動化されることを特徴とするコンピュータ・メモリ。

【請求項19】導入計画オブジェクトが、アプリケーション・プログラムを表す計画前アプリケーション・オブジェクトと、アプリケーション・プログラムを導入すべきワークステーションのグループを表す計画前グループ・オブジェクトとを含むことを特徴とする、請求項18に記載のコンピュータ・メモリ。

【請求項20】導入計画が、アプリケーションの導入用のデータを含む応答ファイル・オブジェクトと、特定のワークステーション用の応答ファイル・オブジェクト・データをカスタマイズするためのデータを含むカスタマイズ・ファイル・オブジェクトとをさらに含むことを特徴とする、請求項19に記載のコンピュータ・メモリ。

【請求項21】妥当性検査手段が、導入計画オブジェクト内のデータとして指定されたターゲット・ワークステーション上のファイル・ディレクトリを検査して、そのファイル・ディレクトリが物理的に有効でネットワーク上でアクセス可能であるかどうかを判定する手段をさらに含み、ターゲット・ワークステーションが、そのアプリケーションを導入すべきワークステーションのグループのうちの1つであることを特徴とする、請求項19に記載のコンピュータ・メモリ。

【請求項22】妥当性検査手段が、ワークステーションのグループ上でアプリケーションの導入に必要なコード・サーバ・ワークステーション上のオブジェクトの存在を確認する手段をさらに含み、コード・サーバ・オブジェクトが、導入計画オブジェクトの子オブジェクトではないことを特徴とする、請求項19に記載のコンピュータ・メモリ。

【請求項23】妥当性検査手段が、アプリケーション・オブジェクトが、ネットワーク内の第1のアプリケーションに更新内容を提供するアプリケーションを表していることを判定する手段と、アプリケーション・コンテナ・オブジェクト内の第1のアプリケーションを表す第1のアプリケーション・オブジェクトの存在を確認する手段とをさらに含むことを特徴とする、請求項19に記載のコンピュータ・メモリ。

【請求項24】導入計画オブジェクトが、導入すべき複

数のアプリケーションと対応する応答ファイルとをそれぞれ表す、複数の計画オブジェクト・オブジェクトと複数の対応する応答ファイル・オブジェクトとを含み、変換ステップが、ワークステーションのグループ内の特定のワークステーションに応じて複数の応答ファイル・オブジェクトをカスタマイズするために、カスタマイズ・ファイル・オブジェクトを複数の応答ファイル・オブジェクトに適用することをさらに含むことを特徴とする、請求項20に記載のコンピュータ・メモリ。

【請求項25】複数の導入計画オブジェクトについて、事前妥当性検査ステップと、妥当性検査ステップと、変換ステップとを繰り返す手段と、各ワークステーションごとにワークステーション待ち行列に複数の導入計画オブジェクトを待ち行列化する手段とをさらに含むことを特徴とする、請求項18に記載のコンピュータ・メモリ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的には、コンピュータ・ネットワークでのソフトウェアの電子配布に関する。より具体的には、本発明はネットワーク上でネットワーク管理者が計画を実行する前にソフトウェアの導入に関する提案済み計画を妥当性検査 (validating) するためのアーキテクチャに関する。

【0002】

【従来の技術】ローカル・エリア・ネットワーク (LAN) や広域ネットワーク (WAN) などの計算機ネットワークを相互接続する複数のデータ処理システムを結合することは一般的なことである。それぞれのコンピュータ・システムは、複数のソフトウェア・プログラムを常駐させることになる。このようなネットワークの初期には、ネットワーク管理者が1組のディスクを備えた各ワークステーションに1人ずつ向かい、導入、構成、または保守が必要なソフトウェア製品ごとに手作業でパラメータを入力していた。わずか数台のワークステーションであれば、複数の管理者とユーザがそれぞれのワークステーションに手作業でディスクを挿入し、導入または構成プロセス中の質問プロンプトに回答することができた。しかし、ネットワークのサイズが拡大するにつれ、当然のことながら、このような方法でソフトウェアを導入することがより難しくなってきた。また、ネットワークのサイズや複雑さ、ならびにコンピュータ・ソフトウェア製品間の相互依存度が増すにつれ、それぞれのワークステーションを構成することがますます複雑で時間を要し、エラーが発生しやすいプロセスになってきた。管理者は、もはやこの手作業による方法を行えなくなり、むしろ、ネットワークを介してワークステーションにアプリケーションを導入し構成するよう特別に設計された製品に転換している。

【0003】したがって、リモートからコンピュータ・

ソフトウェアを導入し、ワークステーションを構成することが、ますます、受け入れられるようになってきた。このプロセスについて一般に使用されている用語は、電子ソフトウェア配布 (ESD) である。ネットワーク内のソフトウェアの電子配布は、ほぼ10年間の間、当技術分野で知られてきた。より一般的なESD製品としては、IBMのNetView分散管理プログラムおよびLAN構成導入分散ユーティリティや、Novell社のNetwork Navigatorなどがある。これらの製品は、ネットワークによるソフトウェアの配布、導入、構成という面倒な作業を軽減するよう設計されている。

【0004】ESD製品はすでに数年間存在しているが、ネットワーク管理者が直面するすべての問題に完全に対処するわけではない。ネットワーク管理者が直面する問題のうち未対応の問題の1つは、ネットワーク内の1群のワークステーション上での複数のソフトウェア・プログラムの配布、導入、構成を計画し実行することである。ソフトウェア製品間の相互依存度が増し、ネットワークが複雑であるため、これは気の重い作業である。

【0005】これまで、ネットワーク管理者の負担を軽減するための努力が行われてきた。これまでの努力は、ネットワークによる画像の伝送方法や、導入および構成プロセスで必要になるファイルの物理的構築など、ネットワーク導入の「配管工事 (plumbing)」に集中していた。これらの設計は、ネットワーク全体でのアプリケーションの伝送などの物理的作業の際に管理者にとって有用であったが、何が物理的作業であるかを計画する際には管理者に役立たなかった。

【0006】本発明は、ネットワーク導入構成計画プロセスを一連の個別オブジェクトに分解するもので、この個別オブジェクトは、問題を抽出し、LAN上の1組のワークステーションに導入し構成する予定のアプリケーションをセットアップし表示させるために管理者が使用する容易でオブジェクト指向の図形手段を提供するのに役立つものである。管理者は、このオブジェクト指向の表現を使用して、物理的な導入構成プロセスに必要なファイルを作成することができる。本発明は、管理者に対して問題を高レベルの表現で表示し、物理的な実施の多くから管理者を保護し、管理者が計画のためのブロックの構築に集中できるようにするものである。この設計を使用すると、管理者は、計画オブジェクトで妥当性検査し、物理的な導入および構成プロセスに必要な実際のファイルを作成するためにコミットすることができる、オブジェクトを「プレイグラウンド」すなわち非計画領域に作成することができる。

【0007】本発明は、電子ソフトウェア配布の重要な改良策の1つを表している。

【0008】

【発明が解決しようとする課題】したがって、本発明の目的は、導入計画オブジェクトを妥当性検査すること

にある。

【0009】本発明の他の目的は、サーバ・ディレクトリなどの前提ファイルを妥当性検査することにある。

【0010】本発明の他の目的は、もともと計画から欠落していた前提オブジェクトまたは属性を導入計画に追加することにある。

【0011】本発明の他の目的は、LAN上のワークステーションでのアプリケーションの導入と構成のために必要なファイルを作成するために導入計画をコミットすることにある。

【0012】本発明の他の目的は、オブジェクト指向ではなく、手続きベースのネットワーク導入ツールと結合することにある。

【0013】

【課題を解決するための手段】上記およびその他の目的は、ネットワークでアプリケーションを導入するために導入計画オブジェクトをコミットすることによって達成される。導入計画オブジェクトは、アプリケーション・プログラムを表す計画オブジェクトと、そのアプリケーション・プログラムを導入するワークステーションのグループを表す計画オブジェクトを含む。コミット・プロセスの一部として、導入計画オブジェクトは、その子オブジェクトを検査し、必要であれば導入計画オブジェクトに追加の子オブジェクトを追加することによって事前妥当性検査され、導入計画オブジェクトとその子オブジェクトのデータにエラーがないかどうか検査することによって妥当性検査され、ネットワーク内でアプリケーションを導入するネットワーク導入エンジンに使用可能なデータ構造に変換される。そのアプリケーションの導入が応答ファイルと、特定のワークステーションに応じて応答ファイル・オブジェクト・データをカスタマイズするためのデータを含むカスタマイズ・ファイル・オブジェクトとを必要とする場合、導入計画は応答ファイル・オブジェクトをさらに含む。

【0014】妥当性検査ステップでは、導入計画オブジェクト内のデータとして指定されたターゲット・ワークステーション上のファイル・ディレクトリを検査して、そのファイル・ディレクトリとファイル・オブジェクトが物理的に有効で、ネットワーク上でアクセス可能であるかどうかを判定する。また、妥当性検査ステップでは、ワークステーションのグループ上でのアプリケーションの導入に必要なオブジェクトがコード・サーバ・ワークステーション上に存在することも確認する。好ましい実施例では、コード・サーバ・オブジェクトは、導入計画オブジェクトの子オブジェクトではない。また、このステップでは、アプリケーション・オブジェクトが、ネットワーク内の第1のアプリケーションへの更新を行うアプリケーションを表すかどうかも判定する。このようなアプリケーションを表す場合、アプリケーション・

コンテナ・オブジェクト内の第1のアプリケーションを表す第1のアプリケーション・オブジェクトが存在することを確認する。また、妥当性検査ステップでは、ターゲット・ワークステーション上のオペレーティング・システムとその訂正サービスのレベルも確認することができる。

【0015】本発明の重要な特徴の1つは、導入計画オブジェクトが通常、複数の計画オブジェクト・オブジェクトと、対応する複数の応答ファイル・オブジェクトとを含むことになる点である。これは、計画内のそれぞれのワークステーションに複数のアプリケーションを導入する予定で、対応する応答ファイルを1つずつ必要とするからである。ワークステーションのグループ内の特定のワークステーションに応じて複数の応答ファイル・オブジェクトをカスタマイズするため、カスタマイズ・ファイル・オブジェクトを複数の応答ファイル・オブジェクトに適用することができる。

【0016】さらに、本発明は、複数の導入計画オブジェクトのために事前妥当性検査、妥当性検査、変換 (transforming) の各ステップを繰り返すことも包含する。複数の導入計画オブジェクトは、それぞれのワークステーションごとに待ち行列化することができる。

【0017】

【発明の実施の形態】本発明は、数種類のオペレーティング・システム下で様々なコンピュータ上またはコンピュータの集合上で動作することができる。このコンピュータとしては、たとえば、パーソナル・コンピュータ、ミニ・コンピュータ、メインフレーム・コンピュータ、他のコンピュータの分散ネットワーク内のコンピュータなどが可能である。特定のコンピュータの選択は、ディスクやディスク記憶域の要件のみに制限されるが、IBM PS/2 (TM) コンピュータ・シリーズのコンピュータは本発明で使用できるはずである。IBMのPS/2コンピュータ・シリーズの詳細については、Technical Reference Manual Personal Systems/2 Model 50, 60 Systems IBM Corporation (部品番号68X2224、資料番号S68X-2224) および Technical Reference 2 Manual Personal Systems/2 (Model 80) IBM Corporation (部品番号68X 2256、資料番号S68X-2254) を参照されたい。IBM PS/2 パーソナル・コンピュータが動作可能なオペレーティング・システムの1つは、IBMのOS/2 2.0 (TM) である。IBMのOS/2 2.0オペレーティング・システムの詳細については、OS/2 2.0 Technical Library, Programming Guide Vol. 1, 2, 3 Version 2.00 (資料番号10G6261、10G6495、10G6494) を参照されたい。

【0018】代替態様のコンピュータ・システムは、AIX (TM) オペレーティング・システム上で動作するIBM RISCシステム/6000 (TM) のコンピュータ系列に含まれるものと考えられる。RISCシス

テム/6000の各種モデルについては、RISC System/6000, 7073 and 7016 POWERstation and POWERserver Hardware Technical reference (資料番号SA23-2644-00)などのIBMの数多くの資料に記載されている。AIXオペレーティング・システムについては、General Concepts and Procedure -- AIX Version 3 for RISC System/6000 (資料番号SC23-2202-00)ならびにIBMのその他の資料に記載されている。

【0019】図1には、システム・ユニット11と、キーボード12と、マウス13と、表示装置14とを含むコンピュータ10がブロック図形式で示されている。システム・ユニット11は、1つまたは複数のシステム・バス21を含み、そのバスに様々な構成要素が結合され、そのバスにより様々な構成要素間の通信が実施される。マイクロプロセッサ22は、システム・バス21に接続され、やはりシステム・バス21に接続された読取り専用メモリ(ROM)23とランダム・アクセス・メモリ(RAM)24によってサポートされている。IBM PS/2シリーズのコンピュータのマイクロプロセッサは、386または486マイクロプロセッサを含むIntelのマイクロプロセッサ・ファミリーの1つである。しかし、この特定のコンピュータでは、68000、68020、68030マイクロプロセッサなどのMotorolaのマイクロプロセッサ・ファミリーや、IBM製またはHewlett Packard、Sun、MotorolaなどのPowerPCチップなどの様々な縮小命令セット・コンピュータ(RISC)マイクロプロセッサを含み、かつこれに限定されないその他のマイクロプロセッサも使用可能である。

【0020】ROM23は、各種コードのうち、対話、ディスク・ドライブ、キーボードなどの基本ハードウェア操作を制御する基本入出力システム(BIOS)を含んでいる。RAM24は、オペレーティング・システムとアプリケーション・プログラムがロードされるメイン・メモリである。メモリ管理チップ25は、システム・バス21に接続され、RAM24とハード・ディスク・ドライブ26およびフロッピー・ディスク・ドライブ27とのデータのやりとりを含む、直接メモリ・アクセス操作を制御する。やはりシステム・バス21に結合されているCD-ROM32は、マルチメディア・プログラムやプレゼンテーションなどの大量のデータを格納するために使用する。

【0021】また、このシステム・バス21には、キーボード制御装置28、マウス制御装置29、ビデオ制御装置30、音声制御装置31という様々な入出力制御装置も接続されている。予想できるように、キーボード制御装置28はキーボード12用のハードウェア・インタフェースを提供し、マウス制御装置29はマウス13用のハードウェア・インタフェースを提供し、ビデオ制御装置30は表示装置14用のハードウェア・インタフェ

ースであり、音声制御装置31はスピーカ15aおよび15b用のハードウェア・インタフェースである。トークン・リング・アダプタなどの入出力制御装置40は、同様に構成された他のデータ処理システムへのネットワーク46による通信を可能にするものである。

【0022】本発明の好ましい実施態様の1つは、前述のように一般的に構成された1つまたは複数のコンピュータ・システムのランダム・アクセス・メモリ24に常駐する命令セット48~52である。コンピュータ・システムが要求するまで、この命令セットは、ハード・ディスク・ドライブ26、またはCD-ROM32で最終的に使用される光ディスクやフロッピー・ディスク・ドライブ27で最終的に使用されるフロッピー・ディスクなどの取外し可能メモリなど、他のコンピュータ・メモリに格納することができる。当業者であれば、命令セットの物理的記憶によって、それが電気的、磁氣的、または化学的に格納される媒体が物理的に変化し、その結果、コンピュータが読取り可能な情報がその媒体に収容されることに留意されたい。命令、記号、文字などの用語で本発明を説明する方が便利であるが、これらの用語ならびに同様の用語はいずれも適切な物理要素に関連するものであることに留意されたい。また、オペレータに関連する可能性のある比較、妥当性検査、その他の用語で本発明を説明する場合も多い。ここに記載し、本発明の一部を形成するいずれの操作もオペレータのアクションは望まれない。これらの操作は、電気信号を処理して他の電気信号を生成する機械による操作を意味する。

【0023】ワークステーションが統合されるネットワークとしては、ローカル・エリア・ネットワーク(LAN)または広域ネットワーク(WAN)があり、後者は既知のコンピュータ・アーキテクチャで動作する他のノードまたは複数システムのネットワークへのテレプロセッシング接続を含む。いずれのノードでも、1つまたは複数の処理システムが存在し、それぞれが多かれ少なかれ前述のように構成された単一ユーザまたは複数ユーザ・システムである可能性がある。これらの処理システムは、サービスを要求する側か提供する側かに応じて、クライアントまたはサーバ・ワークステーションとして動作する。特定の実施態様では、本発明は、LANサーバ48、LAN CIDユーティリティ50、および本発明が実施されているネットワーク導入アプリケーション52を含む、IBM OS/2 LANサーバ・アーキテクチャによって相互接続された複数のIBM互換ワークステーション上で動作する。これらのアプリケーションは、一緒にパッケージ化されている場合もあれば、別々のアプリケーションとして販売される場合もある。ローカル・エリア・ネットワークに関する簡単な説明は、ラリー・E・ジョーダン(Larry E. Jordan)およびブルース・チャーチル(Bruce Churchill)著Communication and Networking For The IBM PC (Robert J. Brady発

行 (A Prentice Hall Company) 1983年) に記載されている。管理者にとって最も単純な構成は、LAN上に1つのコード・サーバを設け、そのワークステーション上でネットワーク導入プログラムを実行する方法であると思われる。この設計は、管理者が複数のコード・サーバをLAN上で定義することを妨げるものではない。

【0024】IBMによって定義されている構成、導入、分散 (CID) の各種方法の1つは、応答ファイルの使用と、製品自体の導入プログラムとの組合せとに基づいている。CIDプロセスでは、ネットワーク管理者はネットワーク内の1台または複数台のマシンを、そこからネットワーク内の他のワークステーションが応答ファイルを含むコンピュータ・ソフトウェア・イメージを受け取るコード・サーバとして選択する。応答ファイルとは、導入または構成あるいはその両方のプロセス中にプログラムによって尋ねられる、対応する1組の質問に対する1組の応答を含むASCIIフラット・ファイルである。一般的なCIDプロセスに関する詳細については、IBMの資料であるLAN Configuration Installation and Distribution Utility Guide (資料番号S10H-9742-00) を参照されたい。リダイレクトされた導入構成を自動化するプログラム・モジュールは、LAN構成導入分配ユーティリティ (LCU) と呼ばれる。以下に示す実施例の説明における詳細の多くは、ネットワーク内でアプリケーションを導入するという実際の物理的作業を実施するためのエンジンとしてLCUを使用することによるものである。以下の好ましい実施例では、本発明のコミット・プロセスによって生成されるファイルは、LCUが必要とするコマンド・ファイルと応答ファイルである。これは、本発明のオブジェクト・ベースの設計を使用すると、LCUなどの手続き型導入エンジンにとって有用なファイルを作成できることを示している。他の導入エンジンを使用すると、コミット・プロセスにより、各種のファイルあるいは導入エンジンがオブジェクト指向であれば各種のオブジェクトが生成されるはずである。LCUコマンド・ファイルは、ネットワーク管理者WksAppIPがワークステーション上での導入を計画している製品ならびにその製品を導入または構成すべき順序を識別する。管理者は、ネットワーク上の各ワークステーションごとに別々のLCUコマンド・ファイルを用意することができる。応答ファイルには、そのワークステーションにソフトウェア製品を導入するのに必要な情報が含まれている。管理者は、アプリケーションごとに応答ファイルを1つずつ用意することができるが、その1つの応答ファイルは、アプリケーションの導入または構成プログラムによってサポートされている場合、追加の組込み応答ファイルを含むことができる。ワークステーション固有の情報が一切なければ、応答ファイルを複数のワークステーション間で共用することもできる。これらは同じ規則によって再導入または構成され

るが、応答ファイルをアプリケーション間で共用することはできない。他に生成されるファイルとしては、コード・サーバの構成に使用する構成ファイルと、コード・ファイルへのアクセス権をクライアントに与えるために使用する権限リスト・ファイルがある。アプリケーション・イメージは、そのアプリケーションの導入元であるコード・サーバ上に格納されている。導入の現行状態を追跡するLAN CIDユーティリティの手順により、各ステップが正しい順序で確実に実行される。当業者であれば、この環境が本発明を実施可能な数多くの環境の1つにすぎないことに留意されたい。

【0025】本発明の重要な態様の1つは、導入計画オブジェクトとその副構成要素が1つのオブジェクト指向システム内で実現されることである。当然のことながら、ここに記載する実施例が示すように、オブジェクト指向の計画オブジェクトは、LAN CIDユーティリティなどの手続きベースの導入エンジンに移植することができる。オブジェクトやオブジェクト指向プログラミングは既知であるが、いくつかの面についてはここで言及する価値がある。それぞれのオブジェクトは、そのデータについて機能する所与のデータ属性とメソッドとを有する。データは、オブジェクトによって「カプセル化」されていると言われ、そのオブジェクトに属すメソッドによってのみ変更することができる。通常、メソッドは、所望のメソッドを識別し、必要な引数を供給するメッセージをオブジェクトに送ることによって呼び出される。新しいクラス・オブジェクトを作成するためにクラス・オブジェクトをサブクラス化することができる。「継承」とは、既存のオブジェクトからメソッドやデータ構造などのすべてのプロパティを継承する既存のオブジェクトから新しいオブジェクトを派生させる能力である。新しいオブジェクトは、既存のクラスの既存のメソッドに追加されるか、またはその既存のメソッドを指定変更する新しいメソッドなど、所与の固有の特徴を有する場合がある。新しいサブクラスでは、その既存の基本クラスからそれを区別するメソッドとデータを指定することだけが必要である。したがって、ソフトウェア開発者は、新しいコード全体を開発する必要はない。開発者は、そのソフトウェアの新しい固有の特徴を指定するだけでよい。オブジェクト指向の技術、概念、規則に関する背景情報については、グレーディー・ブーク (Grady Booch) 著 Object Oriented Design With Applications (The Benjamin/Cummins Publishing Company、1990年) およびB. マイヤー (Meyer) 著 Object Oriented Software Construction (Prentice Hall、1988年) を参照されたい。本発明の場合、導入計画およびその構成要素のオブジェクト指向性により、複数の計画オブジェクトまたはグループ・オブジェクト用のオブジェクトのインスタンスを作成することによって基本アプリケーションおよびワークステーション・オブジェクトの再使用が可

能になる。

【0026】導入計画(plan)オブジェクトと、それに收容される各種オブジェクトの階層表現を図2に示す。オブジェクトは、「プレイグラウンド」内の計画オブジェクトの外部に存在するか、またはそのオブジェクトの「計画中」インスタンスを作成する計画オブジェクトの内部に移動させることができる。計画オブジェクト200は階層の最上部に位置し、管理者は一度に複数の計画オブジェクトを操作することができる。好ましい実施例によれば、有効な計画オブジェクトには、1つまたは複数の計画中アプリケーション(AppIP)オブジェクト202、210、214が收容されることになる。AppIPオブジェクトは、ワークステーションのグループ上に導入または構成されるアプリケーション・プログラムを表している。また、有効な計画オブジェクトには、1つの計画中グループ(GroupIP)オブジェクト220も收容されることになる。計画中グループ・オブジェクトはグループ・オブジェクトに似ているが、計画中ワークステーション・オブジェクトだけを保管し、ワークステーション・オブジェクトは保管しない。すべての計画中ワークステーション・オブジェクトが一義的である限り、1つの計画中グループが単一組のワークステーションまたはネットワーク全体を表すことも可能である。一義性は、ワークステーション識別名から同定される。本発明は、1つの導入計画で複数の計画中グループ・オブジェクトをサポートするように容易に拡張することができる。

【0027】各AppIPオブジェクトは、0または1個のカスタマイズ・ファイル(CustFile)204と、0または1個の計画中カテゴリ(CatIP)オブジェクト206、216、226、236とを收容することができる。また、各計画中アプリケーション・オブジェクトは、導入など、それが実行するアクションを1つ備え、その結果、計画中アプリケーションのアクションと一致する計画中カテゴリ・オブジェクトを1つだけ有することが好ましい。計画中アプリケーション・オブジェクトがアクションを実行するための応答ファイルが必要としない場合、計画中カテゴリも必要としない。CustFileオブジェクトは、垂直軸にキーワード、水平軸にワークステーション名を備えたマトリックス・ファイルとして構成することができる。当業者は、他の構成も可能であることに留意されたい。たとえば、CustFileオブジェクトは、表形式で格納されたインスタンス日付とともに応答ファイルを修正するのに必要なすべてのメソッドを含むことができる。コミット時には、1つまたは複数の計画中アプリケーション・ファイルにそのCustFileオブジェクトを適用し、特定の1組のワークステーションに応じてそれをカスタマイズする。CatIPは、導入または構成など、計画中ワークステーション・オブジェクト上で実行可能な、

AppIP用の指定アクションを表している。各CatIPオブジェクトは、0または1個の応答ファイル(RspFile)208、218、228、238を收容することになる。RspFileは、アプリケーションを導入または構成するときに質問またはパネルへのオペレータ入力の代わりに計画中アプリケーション導入または構成プログラムが使用する、ASCIIキーワード等価値ファイルである。当業者は、GroupIPオブジェクトの下にCustFileオブジェクトも格納するようにこの実施例を拡張可能であることに留意されたい。

【0028】GroupIPオブジェクトは、1つまたは複数の計画中ワークステーション(WksIP)オブジェクト222、240、244を收容することになる。各WksIPオブジェクトは、1つまたは複数の計画中ワークステーション・アプリケーション(WksAppIP)オブジェクト224、230、234を收容することになる。各WksIPオブジェクトごとのWksAppIPオブジェクトの数は、AppIPオブジェクトの数に比例する。1つのワークステーション下の各WksAppIPオブジェクトは、応答ファイルおよびその他の情報を伝播するためのAppIPオブジェクトへのリンクを有する。前述のように、WksAppIPオブジェクトは、AppIPオブジェクトの場合のように、0または1個のCatIPオブジェクトも收容することができる。

【0029】導入計画オブジェクトを作成するネットワーク導入プログラムは、1組の完全定義アプリケーションまたはOS/2 Warp™などのオペレーティング・システムとともに出荷される場合がある。このような構成には、導入および構成を実行するためにアプリケーションが必要とするすべての属性とコマンドが含まれている。管理者が追加のアプリケーションの導入と構成または既存のアプリケーションの導入と構成の修正を必要とする場合、まず、そのネットワーク内のワークステーションのグループと単独ワークステーションとを表す、ワークステーション・グループ・オブジェクトとワークステーション・オブジェクトとを作成し、計画オブジェクトを作成し、そのワークステーション・グループとアプリケーションを新たに作成した計画にドラッグすることから始めることができる。手作業でまたはカスタマイズ・ファイルを使用して応答ファイルを修正する必要がない場合、管理者は、物理的な導入および構成プロセスに必要なファイルを作成するためにただちにその計画をコミットすることができる。様々な計画のためにアプリケーション・オブジェクトとワークステーション・オブジェクトを使用することは、本発明で特に重要な態様の1つである。他の計画で使用するためにこれらのオブジェクトをもう一度作成する必要はない。

【0030】図3および図4は、導入計画内のオブジェ

クトに関連するが、導入計画内に存在しないオブジェクトを示している。図3のコード・サーバ・コンテナ・オブジェクト250は、1つまたは複数のコード・サーバ・オブジェクト252、260、264を含む。コード・サーバは、その他の情報のうち、ワークステーション上に導入または構成するアプリケーションのイメージを格納する。各コード・サーバ・オブジェクト252は、その製品イメージを表す、1つまたは複数のアプリケーション・イメージ (App Image) オブジェクト254および258を有することになる。1つのApp Image オブジェクトは、アプリケーション・コンテナ・オブジェクト内の1つのアプリケーション・オブジェクトへのリンクを有する。リンクとは、ネットワーク導入プログラム内の他のオブジェクトを指し示す、オブジェクトの属性である。

【0031】図4は、1つまたは複数のアプリケーション (App) オブジェクト272、288、292を保管するアプリケーション・コンテナ・オブジェクト270を示している。各App オブジェクトは、1つまたは複数のカテゴリ (Cat) オブジェクトを収容することができる。各Cat オブジェクトは、導入または構成など、App オブジェクトが実行可能なアクションに対応する。各Cat オブジェクトは、1つまたは複数の Rsp File オブジェクトを有することができる。App オブジェクトが計画に追加されると、そのオブジェクトはApp IP オブジェクトに変換され、管理者が選択したアクション・タイプに対応するCat オブジェクトだけがCat IP オブジェクトに変換され、Rsp File がある場合はそれにデフォルトのマークが付けられる。デフォルト応答ファイルは、ほとんどの場合、管理者が導入または構成値を設定するために使用するものになるはずである。

【0032】グラフィカル・ユーザ・インタフェースでは、ツリー・ビュー・ウィンドウ内のアイコンによるなど、同じ視覚的方法でApp オブジェクトとApp IP オブジェクトが管理者に示される。実施のために、様々なオブジェクトが計画オブジェクトと非計画オブジェクトに存在する。計画オブジェクトと非計画オブジェクトを表すために2つの互いに異なるが関連のあるオブジェクトを使用するか、またはそれが計画オブジェクトの一部であるかどうかを判定し、異なる属性を有するかまたは表示するためにそのクラスのxxxxxインスタンスから呼び出せる属性フラグまたはメソッドとともに1つのクラスを使用することもできる。図5は、ネットワーク導入プログラムを提示するために可能なグラフィカル・インタフェースを示している。図5には計画オブジェクトのフォルダを示すオブジェクトのツリー・ビューが示されている。このフォルダは、「Plan Version 1」という計画オブジェクトの内容をほとんど示すようにオープンされている。「Plan Version 1」オブジェクトは、

「OS/2 Warp Version 3」というタイトルのApp IP オブジェクトと、「Install - Category 7」というタイトルのCat IP オブジェクトも含んでいる。また、計画オブジェクトは、「LAN Server 4.0 Entry- Requester」というタイトルの別のApp IP オブジェクトと、「Workstation Group 1」というタイトルのGroup IP オブジェクトも含んでいる。このGroup IP オブジェクトは、「Ted」、「John」、「Barbara」というタイトルのWks IP オブジェクトを示すように展開されている。「Barbara」というWks IP オブジェクトは、「OS/2 Warp Version 3」というタイトルのWks App IP オブジェクトを示すように展開されており、そのオブジェクトは同じタイトルのApp IP オブジェクトと対応する。このWks App IP オブジェクトは、「Install - Category 7」というタイトルのCat IP オブジェクトを示すようにオープンされ、それは「F:\nwi\MODELRSP\OS2W30\INS\WARPSAMP.RSP」というファイル名タイトルを備えた応答ファイルを含んでいる。もう一方のWks App IP オブジェクトである「LAN Server 4.0 Entry - Requester」は図示していないが、これも「Barbara」ワークステーションの一部になるはずである。

【0033】図19の上の図は、ネットワーク導入プログラム用の最上位レベルのコンテナを示し、「Plan Construction」、「Status」、「Code Servers」、「Templates」の各オブジェクトを表示している。「Plan Construction」オブジェクトは、図18のウィンドウに通じている。「Status」オブジェクトは、ワークステーションごとまたは計画ごとにアプリケーションの導入および構成の状況を管理者が表示できるようにするためのダイアログに通じている。「Code Servers」オブジェクトは、オープンすると、ネットワーク導入プログラムで定義されたコード・サーバ・オブジェクトを表示する。

「Templates」オブジェクトは、オープンすると、管理者が計画またはアプリケーション・オブジェクトなどの正規オブジェクトを作成する際に使用するテンプレート・オブジェクトを提供する。オープンした「Templates」オブジェクトは図19の下の図に示す。

【0034】図5の図形表現は、ネットワーク導入プログラム内のオブジェクトを管理者に示すための唯一の方法ではない。たとえば、Wks IPの設定ノートブックで、Wks App IP オブジェクトとその子オブジェクトを表示し、選択することも可能である。両方のオブジェクトが示されるわけではないが、App オブジェクトとApp IP オブジェクトなどの同様のオブジェクトは、管理者が関連アイコンを物理的に変更しない限り、同じアイコン表現を有することになる。当業者であれば、ネットワーク導入プログラム用に代替グラフィカル・ユーザ・インタフェースを数多く設計できるはずである。

【0035】導入計画内のオブジェクトを、導入計画用の物理的導入および構成プロセスが使用可能なファイルに変換するコミット・プロセスは、導入計画メニューからコミット・メニュー項目を選択することなどのアクションによって管理者がこの事象を開始することから始まる。このアクションにより、図7のメソッド入口ブロック400で導入計画オブジェクトにCOMMITメソッドが送られる。コミット・プロセスはステップ405に移行し、そこで導入計画オブジェクトはコミット・プロセス中に検出される状況、警告、エラーの各メッセージを格納するためのエラー・ログ・オブジェクトERRORLOGを作成する。妥当性検査プロセス中に検出した重大エラーを追跡するために、VALERRORカウンタも設定される。

【0036】次に、ステップ410で導入計画オブジェクトにPREVALIDATEメソッドが送られる。このPREVALIDATEメソッドは、導入または構成を正常に行うために必要な場合にオブジェクトまたは属性を計画オブジェクトに追加するもので、詳細については図8に示す。PREVALIDATEが復帰した後、415では、計画内に正しいデータがあるかどうかを検査するVALIDATEメソッドが導入計画オブジェクトに送られる。VALIDATEメソッドについては図9に示す。判断ブロック420では、VALERRORカウンタを検査する。VALERRORカウンタがゼロより大きい場合、処理はステップ445に移行し、そこでコミット・プロセス中に検出した警告およびエラーをERRORLOGが通知する。ログの通知後、COMMITメソッドは復帰ブロック450で終了する。

【0037】VALERRORカウンタがゼロの場合、コミット処理はステップ425に移行し、そこで図10に示すAPPLY CUSTOMIZATION FILESメソッドが導入計画オブジェクトに送られる。APPLY CUSTOMIZATION FILESメソッドから復帰後、導入計画オブジェクトはステップ430で計画出力ファイルを作成する。計画出力ファイルは、図27に詳細を示し、ステップ435で呼び出されるINSTALL COMMAND SCRIPT GENERATIONルーチンへの媒介データとして機能する。INSTALL COMMAND SCRIPT GENERATIONルーチンの復帰後、導入計画は判断ブロック440で、コミット・プロセスの残りの部分で何らかのエラーが検出されたかどうかを検査する。エラーが検出された場合、処理はステップ445に移行する。エラーが一切検出されなかった場合、COMMITメソッドはブロック450で終了する。

【0038】導入計画オブジェクト用のPREVALIDATEメソッドは、図8のメソッド入口ブロック500から始まる。事前妥当性検査は、必要な場合にオブジェクトまたは属性を導入計画に追加するという点が妥当性検査とは異なる。妥当性検査では、プロセス中に後で問題を引き起こす可能性があるため管理者に警告する必要がある値とともに、正しい値または間違っただけの有無だけを検査する。処理はステップ505に移行し、そこでMAINTENA

NCE SYSTEM REQUIRED FLAGとMAINTENANCE SYSTEM EXISTS FLAGの両方が偽に設定される。次のステップ510ではカウンタIを1に設定し、判断ブロック515に移行し、そこでIカウンタが導入計画内のAppIPオブジェクトの数以下であるかどうか確認するためにIカウンタを検査する。この条件が真の場合、処理はステップ535に移行し、そこでカウンタIによって索引付けされたAppIPオブジェクトにIS MAINTENANCE SYSTEM REQUIREDメソッドが送られる。このメソッドの復帰後、処理は判断ブロック540に移行し、真に設定されているかどうかを確認するためにMAINTENANCE SYSTEM EXISTS FLAGを検査する。これが真ではない場合、PREVALIDATEメソッドは残りのAppIPオブジェクトを検査する必要がなく、復帰ブロック550で終了することができる。MAINTENANCE SYSTEM EXISTS FLAGが真に設定されている場合、処理はステップ545に移行し、そこでIカウンタを1だけ増分してから、判断ブロック515に移行する。

【0039】判断ブロック515が偽であり、導入計画内のすべてのAppIPオブジェクトが検査された場合、処理は判断ブロック520に移行し、そこで真に設定されているかどうかを確認するためにMAINTENANCE SYSTEM REQUIRED FLAGを検査する。これが真である場合、PREVALIDATEメソッドは復帰ブロック550で終了することができる。MAINTENANCE SYSTEM REQUIRED FLAGが偽に設定されている場合、処理はステップ525に移行し、そこで導入計画オブジェクトは、TRANSPORTというアプリケーション・タイプおよびMAINTENANCE SYSTEMというアクション・タイプと同じ属性内の項目AppObjectから新しいAppIPオブジェクトを作成する。たとえば、OS/2構成では、ワークステーションのハード・ディスク上にあるOS/2プログラムの最小バージョンが保守システム・アプリケーションになる。この最小OS/2は、OS/2プログラムのプレゼンテーション・マネージャ機能またはワークスペース・シェル機能を含まない。OS/2オペレーティング・システム用の導入アクションは、保守システムを必要とするようなプロセスである。次のステップ530では導入計画に新しいAppIPオブジェクトを追加し、PREVALIDATEメソッドは復帰ブロック550で終了する。

【0040】導入計画オブジェクト用のVALIDATEメソッドは、図9のメソッド入口ブロック600から始まる。処理は判断ブロック605に移行し、そこで導入計画オブジェクトは、導入計画が少なくとも1つのAppIPオブジェクトを含むかどうかを検査する。含まない場合、ステップ630でエラーをログ記録してから、処理が判断ブロック635に移行する。導入計画オブジェクトが少なくとも1つのAppIPオブジェクトを有する場合、ステップ610は、次にカウンタIを1に設定することによって実行する。次に判断ブロック615が続

き、そこでIカウンタが導入計画オブジェクト内のAppIPオブジェクトの数以下であるかどうかを確認するためにIカウンタを検査する。この条件が真である場合、処理はステップ620に移行し、そこでカウンタIによって索引付けされたAppIPオブジェクトにVALIDATEメソッドが送られる。このメソッドの復帰後、処理はステップ625に移行し、そこでIカウンタを1だけ増分してから、判断ブロック615に移行する。

【0041】判断ブロック615が偽であり、導入計画オブジェクト内のすべてのAppIPオブジェクトが図11に示すようにそれぞれの妥当性検査メソッドを実行した場合、処理は判断ブロック635に移行し、そこで計画がGroupIPオブジェクトを含むかどうかを確認するために導入計画オブジェクトが検査する。計画がそれを含まない場合、ステップ645でエラーをログ記録してから、処理は判断ブロック650に移行する。導入計画オブジェクトがGroupIPオブジェクトを有する場合、ステップ640でVALIDATEメソッドがGroupIPオブジェクトに送られ、その妥当性検査メソッドを実行する。

【0042】判断ブロック650では、生成した応答ファイルの格納場所などのローカル・ディレクトリが正しく指定されているかどうか、またそのディレクトリが存在するかどうかを判定するために導入計画オブジェクトがそれ自体を検査する。ローカル・ディレクトリが存在しない場合、ステップ655でエラーがログ記録されるから、ステップ660に移行する。すべてのローカル・ディレクトリが指定され、存在している場合、処理はステップ660に移行し、そこでカウンタIが1に設定される。次に判断ブロック665が続き、そこでIカウンタが導入計画オブジェクト内に指定されたコード・サーバ・オブジェクトの数以下であるかどうかを確認するためにIカウンタを検査する。この条件が真である場合、処理はステップ670に移行し、そこでカウンタIによって索引付けされたコード・サーバ・オブジェクトにVALIDATEメソッドが送られる。このメソッドの復帰後、処理はステップ675に移行し、Iカウンタを1だけ増分してから、処理が判断ブロック665に移行する。判断ブロック665が偽であり、導入計画オブジェクト内のすべてのコード・サーバが検査された場合、VALIDATEメソッド用の処理は復帰ブロック680で完了する。

【0043】導入オブジェクト計画用のAPPLY CUSTOMIZATION FILESメソッドは、図10のメソッド入口ブロック700から始まる。前述のように、このプロセスは、属性のように特定のワークステーション向けのアプリケーション用の応答ファイル内の値を修正するために管理者が使用する。すなわち、それぞれのワークステーションは、それに固有の何らかの値を有することができる。処理はステップ705に移行し、そこでカウンタIが1に設定される。次に判断ブロック710が続き、そこで

Iカウンタが導入計画に指定されたAppIPの数以下であるかどうかを確認するためにIカウンタを検査する。この条件が偽である場合、APPLY CUSTOMIZATION FILESメソッドは復帰ブロック785で完了する。

【0044】判断ブロック710ですべてのAppIPオブジェクトが検査されたわけではない場合、処理はステップ715に移行し、そこでCustFileオブジェクトを有するかどうかを判定するために、カウンタIによって索引付けされたAppIPオブジェクトを検査する。CustFileオブジェクトが検出されない場合、ステップ735ではカウンタIを1だけ増分し、処理は判断ブロック710に移行する。CustFileオブジェクトが検出された場合、ステップ720でCustFileが読み取られ、導入計画オブジェクトはCustFileオブジェクト内の指定のWksIPオブジェクトごとにキーワードと値の辞書を作成する。次に、ステップ725ではカウンタJを1に設定し、判断ブロック730ではカウンタJがCustFileに指定したWksIPオブジェクトの数以下であることを確認するために検査する。この条件が偽である場合、処理はステップ780に移行する。

【0045】CustFileオブジェクトに処理すべきWksIPオブジェクトが依然として存在する場合、ステップ740でカウンタJを1に設定する。次に判断ブロック745では、カウンタKがカウンタJによって索引付けされたWksIPオブジェクトに属すWksAppIPオブジェクトの数以下であることを確認するために検査する。この条件が偽である場合、処理はステップ780に移行し、そこでカウンタJを1だけ増分してから、次に判断ブロック730を実行する。

【0046】カウンタJによって索引付けされたWksIPオブジェクトに属すWksAppIPオブジェクトのすべてが検査されたわけではない場合、判断ブロック750では、カウンタKによって索引付けされたWksApp2pがカウンタIによって索引付けされたAppZpオブジェクトと同じアクション・タイプとアプリケーション・タイプを有するかどうかを確認するために検査する。XXX高速処理はステップ755に移行し、そこでカウンタKを1だけ増分し、判断ブロック745に移行する。フラグが一致である場合、処理はステップ760に移行し、そこでカウンタKによって索引付けされたWksAppIPオブジェクトに属すCatIPオブジェクトに対応するRspFileオブジェクトが取り出される。次に判断ブロック765では、RspFileオブジェクトがMODELタイプのものであるかどうかを確認するために検査する。これがMODELタイプのものである場合、ステップ770でRspFileオブジェクトがGENERATEDタイプのRspFileオブジェクトに変換される。RspFileがすでにGENERATEDタイプのものである場合、または

すでにGENERATEDタイプに変換されている場合、ステップ775では、カウンタJによって索引付けされたWksIPオブジェクト用の辞書内のキーワード用の値に対応するファイル内の値を置換するために、RspFileオブジェクト全体を探索することにより、CustFileの変更内容をRspFileオブジェクトに適用する。その後、プロセスはステップ755に移行するはずである。

【0047】計画中アプリケーション(AppIP)オブジェクト用のVALIDATEメソッドは図11のメソッド入口ブロック800から始まる。処理は判断ブロック805に移行し、そこでAppIPオブジェクトは指定のアクション・タイプ用にコマンド行が指定されているかどうかを確認するためにそのオブジェクト自体を検査する。たとえば、管理者がAppIPオブジェクトのアクション・タイプについてCONFIGUREを選択した場合、AppIPオブジェクトにはCONFIGUREコマンド行ステートメントが指定されていなければならない。コマンド行が指定されていない場合、ステップ810でエラーをログ記録し、処理は判断ブロック815に移行する。

【0048】そのアクション・タイプについてコマンド行が指定されている場合、処理は判断ブロック815に移行し、そこでAppIPオブジェクトはコード・サーバが指定されたかどうかを確認するために検査する。コード・サーバが指定されていない場合、ステップ820でエラーをログ記録し、処理は復帰ブロック830に移行する。コード・サーバが指定されている場合、ステップ825では指定のコード・サーバ・オブジェクトにDOES APPIIMAGE EXIST FOR APPIPメソッドを送る。このメソッドからの復帰はYESまたはNOになる。復帰がNOである場合、ステップ820でエラーをログ記録する。メソッドからの復帰がYESである場合、処理は判断ブロック830に移行する。

【0049】判断ブロック830では、AppIPオブジェクトのアプリケーション・タイプがCORRECTIVE SERVICEであるかどうかを確認するために検査する。訂正サービス・アプリケーションとは、新しいDLLファイルなどの更新モジュールをアプリケーションに適用するものである。更新モジュールは、問題箇所の修正や小規模な機能強化を含むことができる。アプリケーション・タイプがCORRECTIVE SERVICEではない場合、処理は判断ブロック850に移行する。アプリケーション・タイプがCORRECTIVE SERVICEである場合、判断ブロックでは、参照した訂正サービスAppオブジェクトがアプリケーション・コンテナ・オブジェクトに存在するかどうかを確認するために検査する。タイプがCORRECTIVE SERVICEではないAppまたはAppIPオブジェクトと照らし合わせて、訂正サービスAppオブジェクトまたはAppIPオブジェクトが適用される。訂正サービスAppが存在しない場合、ステップ840でエラーをログ記録

し、処理は判断ブロック850に移行する。訂正サービスAppオブジェクトが存在する場合、AppIPオブジェクトは訂正サービスAppオブジェクトにVALIDATEメソッドを送る。メソッドが復帰すると、処理は判断ブロック850に移行する。

【0050】判断ブロック850では、ステップ855のAppIPオブジェクト用にCatIPオブジェクトが存在するかどうかを確認するために検査する。CatIPオブジェクトが存在する場合、AppIPオブジェクトはCatIPオブジェクトにVALIDATEメソッドを送る。VALIDATEメソッドがCatIPオブジェクトから復帰した場合、またはAppIPオブジェクト用にCatIPオブジェクトが存在しなかった場合、処理は判断ブロック860に移行し、そこでAppIPオブジェクトはそれがCustFileオブジェクトを所有するかどうかを確認するために検査する。

【0051】CustFileオブジェクトがAppIPオブジェクトに属する場合、ステップ865でVALIDATEメソッドがCustFileオブジェクトに送られる。VALIDATEメソッドがCustFileから復帰した場合、またはAppIPオブジェクト用にCustFileが存在しなかった場合、AppIPオブジェクト用のVALIDATEメソッド処理は復帰ブロック870で完了する。当業者は、使用する導入エンジンの要件に応じて、計画中アプリケーション・オブジェクト用の妥当性検査メソッドとして数多くの変形態様が存在することに留意されたい。

【0052】AppIPオブジェクト用のIS MAINTENANCE SYSTEM REQUIREDメソッドは図12のメソッド入口ブロック900から始まる。このメソッドはAppIPオブジェクトで必要になる。というのは、保守システム・アプリケーションが計画の一部である必要がある場合に、AppIPオブジェクトは計画オブジェクトに回答する必要があるからである。処理は判断ブロック905に移行し、そこでAppIPオブジェクトは、AppIPオブジェクトの指定のアクション・タイプについて保守システムが要求する属性が設定されているかどうかを確認するためにそのオブジェクト自体を検査する。属性が設定されている場合、ステップ810ではMAINTENANCE SYSTEM REQUIRED FLAGを真に設定し、処理は判断ブロック915に移行する。属性が設定されていない場合、判断ブロック915では次が実行され、ここでAppIPオブジェクトは、このアプリケーション・タイプがTRANSPORTに設定されているかどうか、ならびにそのアクション・タイプがMAINTENANCESYSTEM ONLYに設定されているかどうかを確認するために検査する。両方のタイプが一致する場合、ステップ920ではMAINTENANCE SYSTEM EXISTS FLAGを真に設定し、処理は復帰ブロック925で完了する。両方のタイプが一致しない場合、IS MAINTENANCE SYSTEM REQUIREDメソッドは復帰ブロック92

5で完了する。

【0053】Appオブジェクト用のVALIDATEメソッドは図13のメソッド入口ブロック1000から始まる。AppオブジェクトとAppIPオブジェクト用のVALIDATEメソッドは互いに異なる。というのは、Appオブジェクトは、その子カテゴリ・オブジェクトの妥当性検査や、CustFileオブジェクトのようにAppIPオブジェクトに属すがAppオブジェクトには属することができないオブジェクトの妥当性検査を行う必要がないからである。処理は判断ブロック1005に移行し、そこでAppオブジェクトは、指定のアクション・タイプ用にコマンド行が指定されているかどうかを確認するためにそのオブジェクト自体を検査する。コマンド行が指定されていない場合、ステップ1010でエラーをログ記録し、処理は判断ブロック1015に移行する。

【0054】そのアクション・タイプについてコマンド行が指定されている場合、処理は判断ブロック1015に移行し、そこでAppオブジェクトは、コード・サーバが指定されているかどうかを確認するために検査する。コード・サーバが指定されていない場合、ステップ1020でエラーをログ記録し、処理は復帰ブロック1030に移行する。コード・サーバが指定されている場合、ステップ1025では指定のコード・サーバ・オブジェクトにDOES APPIMAGE EXISTメソッドを送る。このメソッドからの復帰はYESまたはNOになる。復帰がNOである場合、ステップ1020でエラーをログ記録する。メソッドからの復帰がYESである場合、VALIDATEメソッドは復帰ブロック1030で完了する。

【0055】計画中ワークステーション(WksIP)オブジェクト用のVALIDATEメソッドは図14のメソッド入口ブロック1100から始まる。処理は判断ブロック1105に移行し、そこでWksIPオブジェクトは、オペレーティング・システムのAppオブジェクトが指定されているかどうかを確認するためにそのオブジェクト自体を検査する。オペレーティング・システムのAppオブジェクトが指定されている場合は、WksIPオブジェクト上の導入済みオペレーティング・システムを管理者が把握し、そのオペレーティング・システムがそのオブジェクト上に設定されていることを意味し、処理はステップ1110に移行し、そこでオペレーティング・システムのAppオブジェクトにVALIDATEメソッドが送られる。メソッドの復帰後、判断ブロック1115では、WksIPオブジェクト用に訂正サービス・アプリケーションが指定されているかどうかを確認するために検査する。訂正サービス・アプリケーションが指定されている場合、処理はステップ1120に移行し、そこで訂正サービスのAppオブジェクトにVALIDATEメソッドが送られる。VALIDATEメソッドが復帰すると、処理はステップ1125に移行する。訂正サービスのAppオブジェクトが指定されていない場合も、処理はステップ1

125に移行する。

【0056】オペレーティング・システムのAppオブジェクトが指定されていない場合も、処理はステップ1125に移行し、そこでカウンタIが1に設定される。次に判断ブロック1130が続き、そこでIカウンタがWksIPオブジェクト内のWksAppIPオブジェクトの数以下であるかどうかを確認するためにそのカウンタを検査する。この条件が真である場合、処理はステップ1135に移行し、そこでカウンタIによって索引付けされたWksAppIPオブジェクトにVALIDATEメソッドが送られる。このメソッドの復帰後、処理はステップ1140に移行し、そこでIカウンタを1だけ増分してから、判断ブロック1130に移行する。

【0057】WksIPオブジェクトに属するすべてのWksAppIPオブジェクトが検査された場合、VALIDATEメソッドは復帰ブロック1145で完了する。

【0058】WksAppIPオブジェクト用のVALIDATEメソッドは図15のメソッド入口ブロック1200から始まる。処理は判断ブロック1205に移行し、そこでAppIPオブジェクトは、それがCatIPを含むかどうかを確認するためにそのオブジェクト自体を検査する。CatIPオブジェクトが検出されない場合、VALIDATEメソッドは復帰ブロック1220で終了する。CatIPオブジェクトが検出された場合、WksAppIPオブジェクトはステップ1210でCatIPオブジェクトにVALIDATEメソッドを送る。メソッドの復帰後、処理は復帰ブロック1220で終了する。

【0059】CatIPオブジェクト用のVALIDATEメソッドは図16のメソッド入口ブロック1300から始まる。処理は判断ブロック1305に移行し、そこでCatIPオブジェクトは、それがRspFileオブジェクトを含むかどうかを確認するためにそのオブジェクト自体を検査する。RspFileオブジェクトが検出されない場合、VALIDATEメソッドは復帰ブロック1320で終了する。RspFileオブジェクトが検出された場合、CatIPオブジェクトはステップ1310でRspFileオブジェクトにVALIDATEメソッドを送る。メソッドの復帰後、処理は復帰ブロック1320で終了する。

【0060】RspFileオブジェクトとCustFileオブジェクト用のVALIDATEメソッドは図17のメソッド入口ブロック1400から始まる。処理は判断ブロック1405に移行し、そこでそのファイル・オブジェクトは、その実ファイル名がローカルにアクセス可能なドライブ上のファイルとして物理的に存在するかどうかを確認するために検査する。ローカルまたはネットワークでアクセス可能なドライブ上でそのファイル名を検出できない場合、ステップ1410でエラーをログ記録し、処理は復帰ブロック1415に移行する。ファイル名が突き止められた場合、VALIDATEメソッドは復帰ブロ

ック1415で終了する。

【0061】計画グループ(Group IP)オブジェクト用のVALIDATEメソッドは図18のメソッド入口ブロック1500から始まる。処理はステップ1510に移行し、そこでカウンタIを1に設定する。次に、判断ブロック1515では、IカウンタがGroup IPオブジェクト内のWks IPオブジェクトの数以下であるかどうかを確認するために検査する。この条件が真である場合、処理はステップ1515に移行し、そこでGroup IPオブジェクトはカウンタIによって索引付けされたWks IPオブジェクトにVALIDATEメソッドを送る。このメソッドの復帰後、処理はステップ1520に移行し、そこでカウンタIを1だけ増分し、制御は判断ブロック1510に戻る。判断ブロック1510が偽であって、Group IPオブジェクト内のすべてのWks IPオブジェクトが検査された場合、VALIDATEメソッドは復帰ブロック1525で終了する。

【0062】コード・サーバ・オブジェクト用のVALIDATEメソッドは図19のメソッド入口ブロック1600から始まる。処理は判断ブロック1605に移行し、そこでコード・サーバ・オブジェクトが実際に定義されており、別のオブジェクトで参照されるだけではないかどうかを確認するためにコード・サーバ・コンテナを検査する。コード・サーバが検出されない場合、ステップ1610でエラーをログ記録し、メソッドは復帰ブロック1615で完了する。

【0063】コード・サーバ・オブジェクト用のDOES APPIMAGE EXIST FOR APPIPメソッドは図20のメソッド入口ブロック1630から始まる。処理はステップ1635に移行し、そこでカウンタIを1に設定する。次に、判断ブロック1640では、Iカウンタがコード・サーバ内のアプリケーション・イメージの数以下であるかどうかを確認するために検査する。この条件が真である場合、処理はステップ1645に移行し、そこでアプリケーション・イメージに属性として指定されたアプリケーション・インスタンス名をApp IPのインスタンス名と照らし合わせて検査する。2つのインスタンス名が一致する場合、DOES APPIMAGE EXIST FOR APPIPメソッドは終了し、YES (Y)を返す。2つの名前が一致しない場合、ステップ1650でカウンタIを1だけ増分し、処理は判断ブロック1640に移行する。判断ブロック1640が偽であって、コード・サーバ内のすべてのアプリケーション・イメージが検査された場合、DOES APPIMAGE EXIST FOR APPIPメソッドは終了し、NO (N)を返す。

【0064】INSTALL COMMAND SCRIPT GENERATIONルーチンは、導入計画オブジェクトが作成した計画出力ファイル(POFファイル)を読み取る。計画出力ファイルとその各セクションについては図21に示す。計画セクション2002は、応答ファイル、ログ・ファイル、ユ

ーティリティ・ファイルに使用する計画名およびサーバ名など、その計画に固有のデータを含む。また、計画セクションは、1つのシーケンス・セクション2004と、1つまたは複数のAppセクション2006、ワークステーション・セクション2012、サーバ・セクション2018をも含む。シーケンス・セクション2004は、ワークステーション上での計画実行中に実行する順序でそれぞれのアプリケーション/アクションをリストする。Appセクション2006は、短縮名、アクション、コマンド行、イメージ・サーバと別名、必要な環境など、計画内の各アプリケーションに固有のデータを含む。ワークステーション・セクション2012は、ワークステーション名、ブート・ドライブ、オペレーティング・システムの短縮名など、各ワークステーションに固有のデータを含む。その計画で参照される各サーバごとに、サーバ・セクションが1つずつ存在する。サーバ・セクション2022は、サーバ名と、応答ファイル、ログ・ファイル、ユーティリティ・ファイル用のその別名(マウント点)とを含む。導入計画オブジェクト・コードは数多くの様々な導入エンジンのために移植性が高くなっているため、単に導入エンジン用のファイルを直接生成するのではなく、計画出力ファイルを用意する方が便利である。以下に記載するコマンド・スクリプト生成ルーチンのみ、書き直す必要がある。

【0065】INSTALL COMMAND SCRIPT GENERATIONルーチンは、計画内のワークステーション上で実行されるときに、グラフィカル・ユーザ・インタフェースで計画の進捗状況を示すためのベースとして機能する計画状況ファイルを作成する。計画状況ファイルとその各セクションについては図22に示す。計画セクション2032はその計画に固有のデータを含む。ワークステーション・セクション2034は、計画内の各ワークステーションごとに1つずつ含まれる。各ワークステーション・セクションは、計画内の各アプリケーション・コマンド行に示される各ログ・ファイルごとにワークステーション・ログ2036を1つずつ含むことになる。また、各ワークステーション・セクションは、ワークステーション上の計画の全体的状況、たとえば、進行中、成功、失敗を含むファイルを指し示すワークステーション状況ファイル2042も1つずつ含むことになる。

【0066】INSTALL COMMAND SCRIPT GENERATIONルーチンは、計画内の各ワークステーションごとに4つのファイルも作成する。この4つのファイルは、図23に示すワークステーション待ち行列ファイルと、図24のPREコマンド・ファイルと、図25に示すPOSTコマンド・ファイルと、図26に示すLCUコマンド・ファイルである。

【0067】図23のワークステーション待ち行列ファイルは、ワークステーション上での実行を待っている計画2062の名前から構成される。このファイルは、タ

ターゲット・ワークステーション側で実行される複数の計画用の待ち行列化機構である別のプログラムNWICheckによって使用される。ターゲット・ワークステーション上で計画を実行する際に、このプログラムは待ち行列の最上部から実行済みの計画を除去する。この待ち行列化機構は、アプリケーション・プログラムをネットワークで導入する際の従来の労力に比べ、本発明が優れている重要な点の1つである。複数の計画オブジェクトを順次実行できるようにすることによって、複数のワークステーションでの複数のアプリケーションの導入を編成する作業を分けし、その結果、かなり容易にすることができる。従来の労力では、除去しなければならぬ導入システムを一度に1つしか活動状態にすることができなかった。

【0068】図24に示すPREコマンド・ファイルは、ワークステーションがブート・ディスクから計画実行を開始しない場合にワークステーションが計画を実行できるようにするためのものである。このPREコマンド・ファイルは、標準のコマンド言語で作成されている。これは、1つまたは複数の接続導入ステートメント2072と、1つのネットワーク導入使用可能性ステートメント2078を含む。接続導入ステートメント2072は、計画実行中に必要になるリポート間に活動状態になるワークステーションにリダイレクトされたドライブを追加する。ネットワーク導入使用可能性ステートメント2078は、ワークステーションのリポート間に計画実行を容易にするような、必要なファイルまたはコマンドあるいはその両方を導入する。図25に示すPOSTコマンド・ファイルは、リダイレクトされたドライブと、計画実行中またはPREコマンド・ファイルによってワークステーションに導入されたネットワーク導入使用可能性プログラムとを除去するためのものである。接続除去ステートメント2088は、リダイレクトされたドライブを除去する。ネットワーク導入使用可能性ステートメント2078は、ネットワーク導入使用可能性プログラムによって事前に導入されたファイルまたはコマンドあるいはその両方を除去する。

【0069】図26に示すLCUコマンド・ファイルは、LAN CIDユーティリティ・プログラムが必要とする構文で作成されたREXXコマンド・ファイルであり、そのプログラムが必要とする変数を使用する。INSTALL COMMAND SCRIPT GENERATIONルーチンは、必要なリボートの回数を最小限に保ちながら、複数のフェーズで複数製品の導入を実行するのに必要なREXXコードを動的に作成する。このコードは、ブート・ディスクまたは他のハード・ディスク・ベースの方法により計画実行を開始するワークステーションを処理するために生成される。LCUコマンド・ファイルには7つのセクションがある。セットアップ・セクション2092は変数の初期設定から構成される。基本LCU手続きセクシ

ョン2104は、マシンのリポートや、CID戻りコードの処理などの事柄に使用する、様々なREXX手続きから構成される。INSTALL COMMAND SCRIPT GENERATIONルーチンが作成するREXXコードは接続セクション2094から始まる。接続セクション2094は、ワークステーション上で計画を実行するのに必要な、リダイレクトされたすべてのドライブ用のリダイレクト・タイプ固有の接続（マウント）ステートメントを含む。アプリケーション・イメージ、ログ・ファイル、応答ファイル、その他の事柄すべてについて、接続が存在するものと思われる。変数セクション2096は、ブート・ドライブなど、ワークステーションに固有のいくつかの変数をセット・アップする。製品データ・セクション2098は、LAN CIDユーティリティが必要とする構文で作成された、計画内のアプリケーションごとのステートメントを含む。これらのステートメントとしては、とりわけ、アプリケーションおよびデフォルト応答ファイル用のコマンド行を含むものと思われる。導入セクション2100は、アプリケーションの順序付け、エラー・アクション、リポート情報を含む。CheckIT手続き2102は、エラー状態が発生したときと、導入の終了時に、導入セクション内から呼び出されるネットワーク導入手続きである。

【0070】INSTALL COMMAND SCRIPT GENERATIONルーチンは図27のブロック2200から始まる。ステップ2205では、計画出力ファイル（POF）ファイル名やログ・ファイル名などのデータがないかどうか、コマンド行が解析される。処理はステップ2210に移行し、そこでコマンド行で渡されたPOFファイルが読み込まれ、導入計画で参照された各コード・サーバごとにサーバ情報などの必要データがないかどうか解析される。POFファイル構造については図21に示す。

【0071】次のステップ2215では、ターゲット・ディレクトリで計画が必要とするディレクトリ構造を作成する。このディレクトリ構造は、ワークステーション上で計画を実行するのに使用するツールによって予測される。当業者は、このディレクトリ構造が以下の好ましい実施例で記載するものとはかなり異なる可能性があることに留意されたい。ターゲット・ディレクトリには、その計画用のディレクトリが作成される。この計画ディレクトリから、その計画内の各アプリケーションごとにLCUコマンド・ファイルと、PREコマンド・ファイルと、POSTコマンド・ファイルと、ネットワーク導入状況ファイルとを含む、ログ・ファイル用のディレクトリが作成される。これらのファイルのそれぞれについては、LAN CIDユーティリティを使用して別々のディレクトリを作成しなければならない。というのは、これらのファイルはワークステーションによって命名され、同じディレクトリに同じ名前の2つの個別ファイルが常駐することはできないからである。

【0072】処理はステップ2220に移行し、そこで計画状況ファイルの計画セクションが書き込まれる。計画状況ファイルの構造については図22に示す。次にステップ2225では、サーバ/別名ドライブ・テーブルが構築される。このテーブルは、ターゲット・ワークステーションで計画を実行する際に使用するサーバとそれに対応する別名(マウント点)のすべてを含む。サーバ上のマウント点とは、ドライブ文字としてアプリケーションを接続できる論理位置である。サーバ/別名のすべての組合せをテーブルに追加した後、ステップ2230で重複部分が除去され、固有のサーバ/別名の組合せにドライブが割り当てられる。ドライブZ:は、ネットワーク導入ユーティリティの別名になるドライブに割り当てられる。ドライブY:は、オペレーティング・システム固有のファイルに接続されるドライブに割り当てられる。固有のサーバ/別名の組合せの残りには、ドライブW:から逆の英字順にドライブ文字が割り当てられる。他のデバイス割当ても可能であるが、ターゲット・ワークステーションに物理的に接続されたデバイスが使用する既存のドライブ文字との競合を避けるために、ドライブが逆の英字順に割り当てられる。

【0073】ステップ2240では、計画の実行中に導入すべきアプリケーションの数が決定される。これは、POFファイルで定義されたアプリケーションの数に2を加えたものとして定義される。ネットワーク導入使用可能性プログラム用に1を加え、何らかの場合に導入される一時リダイレクト用に1を加える。

【0074】次に、INSTALL COMMAND SCRIPT GENERATIONルーチンは、POFファイルで定義されたすべてのワークステーションについて、ステップ2245~2280のループになる。ループ内の第1のステップ2250では、現行計画の名前をワークステーションの待ち行列ファイルに付加する。ワークステーション待ち行列ファイルの構造については図23に示す。次のステップ2255では、アプリケーションのコマンド行で変数置換を実行する。ソース・ディレクトリ、ログ・ファイル、クライアント名などのデータはそれぞれの計画またはワークステーションあるいはその両方に固有のものなので、システムに対してアプリケーションを定義するときに管理者が修正するのではなく、計画コミット時に置換しなければならない。次にステップ2260は、計画状況ファイル上の現行ワークステーション用のワークステーション・セクションである。計画状況ファイルについては図22に示す。各アプリケーション・コマンド行上の/1:パラメータによって指定された各ログ・ファイルとともに、ワークステーションの状況ファイルのサーバに対するローカル経路が計画状況ファイルにリストされる。

【0075】次に、ステップ2265では、プロセスはワークステーションのPREコマンド・ファイルを作成

する。PREコマンド・ファイルの構造については図24に示す。ステップ2230で決定したネットワーク導入ユーティリティの別名用として、接続導入使用可能性ステートメントが追加される。接続導入ステートメントとは、オペレーティング・システムのスタートアップ・ファイルと、ファイル・サーバへの接続ステートメントにステートメントを導入するプログラムである。また、必要であれば、オペレーティング・システム固有のファイル用としても接続導入ステートメントが追加される。また、現行ワークステーション用の適切なネットワーク導入ステートメントも追加される。ステップ2270では、POSTコマンド・ファイルが作成される。POSTコマンド・ファイルの構造については図25に示す。PREコマンド・ファイル(またはLCUコマンド・ファイル内の等価物)で実行したことを基本的に「取り消す」ためにステートメントが追加される。

【0076】次にINSTALL COMMAND SCRIPT GENERATIONルーチンはステップ2275でCREATE LCU COMMAND FILEルーチンを呼び出す。このCREATE LCU COMMAND FILEルーチンについては図28に示す。次のステップでは、次のワークステーションに進んで、2245の判断ブロックに移行する。処理すべき追加のワークステーションがまだ存在する場合はステップ2250~2280を繰り返す、存在しない場合はステップ2285でINSTALL COMMAND SCRIPT GENERATIONルーチンが計画状況ファイルをクローズし、呼出し側プログラムに戻る。

【0077】CREATE LCU COMMAND FILEルーチンは図28のブロック2350から始まる。LCUコマンド・ファイルの構造については図26に示す。LCUコマンド・ファイルを作成する際の第1のステップは、ステップ2355で、LCUコマンド・ファイル用の事前定義済みセットアップ・セクションを書き出すことである。セットアップ・セクションはLAN C I Dユーティリティに固有のものであるが、他の導入エンジンを使用した場合でも、同様のセクションがこれらのファイルに共通のものになる可能性がある。ステップ2360では、INSTALL COMMAND SCRIPT GENERATIONルーチンのステップ2230で決定した固有のサーバ/別名対ごとのドライブ用にネットワーク固有の接続ステートメントが書き込まれる。ステップ2365~2380では、CREATE LCU COMMAND FILEルーチンが、WRITE VARIABLES SECTIONルーチン(図29に示す)と、WRITE PRODUCT DATA SECTIONルーチン(図30に示す)と、WRITE INSTALL SECTIONルーチン(図32に示す)と、WRITE CHECKIT PROCEDUREルーチン(図31に示す)とを呼び出す。ステップ2385では、LCUコマンド・ファイルのために上記のステップ2355でコピーした事前定義済み基本手続きがワークステーションのLCUコマンド・ファイルに書き込まれる。次に、CREATE LCU COMMAND FILEルーチンは呼出し側の手続きに戻る。

【0078】WRITE VARIABLES SECTIONルーチンは図29のステップ2450から始まる。このルーチンは、ワークステーションに固有の変数を書き出す。まず、ステップ2455で、プロセスはワークステーションのブート・ドライブに対応するREXX変数を書き込む。REXX構文では、これは、たとえばbootdrive=D:となるはずである。次に、ステップ2460では、ワークステーションでの計画実行中にLCUが使用する事前定義済み変数がこのファイルに書き込まれる。次に、WRITE VARIABLES SECTIONルーチンは、復帰ブロック2465で呼出し側手続きに戻る。

【0079】WRITE PRODUCT DATA SECTIONルーチンは図30のステップ2500から始まる。製品データ・セクションの各構成要素については図26のボックス2098に説明されている。ステップ2505では、ネットワーク導入ユーティリティへの接続に必要な製品データがこのセクションに書き込まれる。これは、INSTALL COMMAND SCRIPT GENERATIONルーチンのステップ2260で説明したように、PREコマンド・ファイルに書き込まれたものと同じになるはずである。必要であれば、オペレーティング・システム固有のファイルへの接続に必要な製品データもこのステップで書き込まれるはずである。ステップ2510では、ネットワーク導入使用可能性プログラム用の製品データが書き込まれる。ネットワーク導入ユーティリティは導入システムを実行するもので、ネットワーク導入使用可能性プログラムはネットワーク導入ユーティリティを導入するものである。ステップ2515～2525のループでは、計画で導入すべき各アプリケーションごとに製品データを書き出す。すべての製品データが書き出された後で、コマンド・ファイルにいくつの製品データ定義が含まれているかを示す1行が書き出される。この数は、INSTALL COMMAND SCRIPT GENERATIONルーチンのステップ2240で決定したものである。次に、WRITE PRODUCT DATA SECTIONルーチンは呼出し側手続きに戻る。

【0080】WRITE CHECKIT PROCEDUREルーチンは図31のステップ2600から始まる。CheckIt手続きは、LCU REXXコマンド・ファイルに追加される手続きである。CheckIt手続きは、アプリケーションの導入が失敗に終わるか、計画が正常に完了したときに必ず呼び出されるものである。CheckIt手続きは、別のプログラムを呼び出して、それにワークステーションの名前と計画の状況（成功または失敗）を渡す。この呼び出されるプログラムNWICHECKは、計画実行中にワークステーションの待ち行列ファイルと状況ファイルの保守を行う。まず、ステップ2605では、CheckIt手続きの事前定義済みの開始が書き込まれる。次に、ステップ2610で、NWICHECKコマンド行での変数置換が行われ、ワークステーション名などのデータとログ・ファイルを置換する。ステップ2610で生

成したNWICHECKステートメントはステップ2615でコマンド・ファイルに書き込まれる。ステップ2620でCheckIt手続きの残りが付加される。次に、WRITE CHECKIT PROCEDUREルーチンは呼出し側手続きに戻る。

【0081】WRITE INSTALL SECTIONルーチンは図32のステップ2700から始まる。導入セクションの各構成要素については図26のボックス2100に説明されている。第1のステップ2705では、状態を0に設定する。次に、ステップ2710では、REXXを書き出すことによりループが構築される。次に、WRITE INSTALL SECTIONルーチンは、判断ブロック2715から計画内のすべてのアプリケーションを通るループになる。

【0082】判断ブロック2715では、これが処理中の第1のアプリケーションであるかどうかをテストし、第1のアプリケーションではない場合はステップ2720で状態が1だけ増分され、第1のアプリケーションである場合は処理が判断ブロック2725に移行する。

【0083】判断ブロック2725では、アプリケーションが保守システムを必要とするかどうかを検査する。必要とする場合、処理はステップ2760に移行し、そこでMAINT SECTIONルーチン（図37）が呼び出される。ステップ2765で状態が1だけ増分され、次にステップ2770でWRITE APPルーチン（図33）が呼び出される。次に、処理は判断ブロック2745に移行する。

【0084】判断ブロック2725で保守システムが必要ではなかった場合、処理は判断ブロック2730に移行し、そこで導入中にプレゼンテーション・マネージャ（PM）が活動状態になることをそのアプリケーションが要求しているかどうかを検査される。プレゼンテーション・マネージャは、グラフィカル・ユーザ・インタフェース自体を管理する必要なしに、アプリケーションがユーザにグラフィカル・ユーザ・インタフェースを提示できるようにするものである。PMが必要な場合、処理はステップ2775に移行し、そこでHARD DISKルーチン（図39）が呼び出される。次に、ステップ2780で状態が1だけ増分され、処理は判断ブロック2745に移行する。判断ブロック2730でPMが必要ではなかった場合、処理はステップ2735に移行し、そこでWRITE APPルーチン（図33）が呼び出される。処理は判断ブロック2740に移行し、そこでこれが処理中の第1のアプリケーションである場合、ステップ2785でBOOT DISKルーチン（図38）が呼び出され、そうでない場合、処理は判断ブロック2745に移行する。

【0085】判断ブロック2745で最後のアプリケーションが処理中ではない場合、処理はステップ2750に移行し、そこでEND SECTIONルーチン（図36）が呼び出される。次に、処理は判断ブロック2715に戻る。判断ブロック2745で最後のアプリケーションが

処理中である場合、処理はステップ2785に移行し、そこでLAST APPルーチンが呼び出される。次に、WRITE INSTALL SECTIONルーチンは呼出し側手続きに戻る。

【0086】WRITE APPルーチンは図33のステップ2850から始まる。このルーチンは、コマンド・ファイル内の1つの全体状態について1つまたは複数のRun Installステートメントを書き出す。ステップ2855では、選択ステートメントの適当なセクションを示すREXX構文が書き出される。次に、ステップ2860では、処理中のアプリケーションについてWRITE RUNINSTALLルーチン（図34）が呼び出される。このルーチンは、コマンド・ファイル内の1つのRun Installステートメントの論理を書き出す。処理は判断ブロック2865に移行し、そこで処理中のアプリケーションがオペレーティング・システムではない場合、ルーチンは呼出し側手続きに戻る。オペレーティング・システムである場合、処理はステップ2870に移行し、そこで計画で定義されたトランスポート・アプリケーション用にWRITE RUNINSTALLルーチンが呼び出される。次に、ステップ2875では、計画内のリダイレクト用にWRITE RUNINSTALLルーチンが呼び出される。ステップ2880では、ネットワーク導入使用可能性プログラム用にWRITE RUNINSTALLルーチンが呼び出される。次に、復帰ブロック2885でWRITE APPルーチンは呼出し側手続きに戻る。

【0087】WRITE RUNINSTALLルーチンは図34のステップ2900から始まる。ステップ2905と2910では、REXX構文が次のようにコマンド・ファイルに書き出される。

【0088】

```

If RunInstall(x) == BAD_RC then
  Call Checkit('STOP')

```

【0089】この場合、xはこのルーチンの呼出しの対象となるアプリケーションのアプリケーション短縮名である。次に、WRITE RUNINSTALLルーチンは呼出し側手続きに戻る。

【0090】LAST APPルーチンは図35のステップ2950から始まる。このルーチンは、LCUコマンド・ファイルの導入セクションを終了するために使用するREXXロジックを書き込む。ステップ2955では、REXX構文が次のように書き込まれる。

【0091】

```

If CALL_AGAIN = 0 then do
  CheckIt('SWITCH')
  call reboot
end
else
  call checkboot
end
otherwise nop
end
end

```

【0092】このコードは、LCU変数であるCALL-AGAINが設定されているかどうかを検査するもので、この変数が設定されていない場合は、（1つ存在すれば）ワークステーション待ち行列内の次の計画に切り替えるためにCheckITルーチンが呼び出される。現行の全体状態になっているいずれかのプログラムがreboot-and-Callback戻りコードを返す場合は、CALL-AGAIN変数がLCUによって1に設定される。call-againが1に設定されている場合、ワークステーションをリポートするためのLCU手続きが呼び出される。次に、LAST APPルーチンは呼出し側手続きに戻る。

【0093】END SECTIONルーチンは図36のステップ3000から始まる。このルーチンはLCUコマンド・ファイル内の1つの全体状態を終了するために使用するREXXロジックを書き込む。ステップ3005と3010では、REXX構文が次のように書き出される。

【0094】

```

call checkboot
end

```

【0095】このコードは、ワークステーションの条件付きリポートのためのLCU手続きを呼び出す。次に、END SECTIONルーチンは呼出し側手続きに戻る。

【0096】MAINT SECTIONルーチンは図37のステップ3100から始まる。このルーチンは、ターゲット・ワークステーション上に保守システムを導入するのに必要なREXXロジックとRun Installステートメントを書き込む。ステップ3105では、選択ステートメントの適当なセクションを示すREXX構文が書き出される。次にステップ3110で、ブート・ドライブがディスクットであるかどうかを検査するための構文が書き出される。ステップ3115では、計画で定義されたオペレーティング・システムの保守バージョン用にWRITE RUNINSTALLルーチンが呼び出される。ステップ3120では、計画で定義されたトランスポート・アプリケーションの保守バージョン用にWRITE RUNINSTALLルーチンが呼び出される。ステップ3125では計画内のリダイレクト用にWRITE RUNINSTALLルーチンが呼び出され、ステップ3130ではネットワーク導入使用可能性プログラム用にWRITE RUNINSTALLルーチンが呼び出される。次に、ステップ3135では、END SECTIONルーチンが

呼び出される。次に、MAINT SECTIONルーチンは呼出し側手続きに戻る。

【0097】BOOT DISKルーチンは図38のステップ3200から始まる。このルーチンは、ブート・ディスクレットから導入を開始したときにターゲット・ワークステーションのハードファイルにネットワーク導入ユーティリティと必須リダイレクタを導入するのに必要なREXXロジックとrunInstallステートメントを書き込む。第1のステップは、これが最後のアプリケーションであるかどうかを判定する判断ブロック3205である。これが最後のアプリケーションである場合、LCUコマンド・ファイルに次のようにREXX構文を書き出してから、ステップ3215に移行する。

```
【0098】if CALL_AGAIN = 1 then do
```

```
【0099】これが最後のアプリケーションである場合はCall-AGAIN変数が検査される。というのは、このアプリケーションがコール・バックを必要としない場合、ハード・ファイル上にコール・バック・ユーティリティを乗せる必要がないからである。この検査は本質的にはリブートを節約する。これが最後のアプリケーションではない場合、処理はステップ3215に移行し、そこでブート・ドライブがディスクレットであるかどうかを検査するための構文が書き出される。ステップ3220では計画内のリダイレクタ用にWRITE RUNINSTALLルーチンが呼び出され、ステップ3225ではネットワーク導入使用可能性プログラム用にWRITE RUNINSTALLルーチンが呼び出される。ステップ3230で終了が書き込まれ、処理は判断ブロック3235に移行し、そこでこれが最後のアプリケーションであるかどうかを検査する。これが最後のアプリケーションではない場合、BOOT DISKルーチンは呼出し側手続きに戻り、最後のアプリケーションである場合は、ステップ3240でREXX構文が次のように書き込まれてから、呼出し側手続きに戻る。
```

```
Do forever
  Select
    when OVERALL_STATE = 0 then do
      if RunInstall(transport) == BAD_RC then
        call CheckIt('STOP')
      if IsBootDriveRemovable = 1 then do
        if RunInstall(redirector) == BAD_RC then
          call CheckIt('STOP')
        if RunInstall(ni_enable) == BAD_RC then
          call CheckIt('STOP')
        end
        call checkboot
      end
    when OVERALL_STATE = 1 then do
      if RunInstall(os_maint) == BAD_RC then
        call CheckIt('STOP')
      if RunInstall(xport_maint) == BAD_RC then
```

```
【0100】
```

```
call checkboot
end
```

【0101】このコードは、現行の全体状態で事前に導入されたプログラムからのCID戻りコードに基づいて、マシンの条件付きリブートを行うためのLCU手続きを呼び出すものである。

【0102】HARD DISKルーチンは図39のステップ3300から始まる。このルーチンは、導入中のアプリケーションがPMを必要とする場合にブート・ディスクレットからブートしたときに、ターゲット・ワークステーションのハード・ファイルにネットワーク導入ユーティリティと必須リダイレクタを導入するのに必要なREXXロジックとRunInstallステートメントを書き込む。ステップ3305では、選択ステートメントの適当なセクションを示すREXX構文が書き出される。次に、ステップ3310では、ブート・ドライブが固定ドライブであるかどうかを検査するための構文がLCUコマンド・ファイルに書き出される。ステップ3315では、計画内のリダイレクタ用にWRITE RUNINSTALLルーチンが呼び出される。ステップ3320では、ネットワーク導入使用可能性プログラム用にWRITE RUNINSTALLルーチンが呼び出される。ステップ3325ではEND SECTIONルーチンを呼び出す。次に、HARD DISKルーチンは呼出し側手続きに戻る。保守システムまたはプレゼンテーション・マネージャを必要としないトランスポートと、保守システムを必要とするオペレーティング・システム・サービスと、プレゼンテーション・マネージャを必要とするアプリケーションを順番に導入する計画によってWRITE INSTALL SECTIONルーチンを実行後、次のような導入セクションで終わるはずである。

```
【0103】
```

```

call CheckIt('STOP')
if RunInstall(redirector) == BAD_RC then
  call CheckIt('STOP')
if RunInstall(ni_enable) == BAD_RC then
  call CheckIt('STOP')
call checkboot
end
when OVERALL_STATE = 2 then do
  if RunInstall(os_service) == BAD_RC then
    call CheckIt('STOP')
  call checkboot
end
when OVERALL_STATE = 3 then do
if RunInstall(app) == BAD_RC then
  call CheckIt('STOP')
if CALL_AGAIN = 0 then do
  call CheckIt('SWITCH')
  call Reboot
end
else call checkboot
end
otherwise nop
end

```

【0104】導入中にINSTALL COMMAND SCRIPT GENERATIONルーチンが生成したファイルを使用する方法の例を以下に示す。NWICHECKは、ターゲット・ワークステーション上での計画実行の順序を制御するプログラムである。

【0105】まず、NWICHECKが実行されると、このプログラムはワークステーション用の待ち行列ファイルを検査する。現行計画は待ち行列内の最初の計画になる。この計画はこの時点では待ち行列から除去されない。これは、ワークステーション上で計画が完了した後で行われる。この点からのPRE、POST、LCUの各コマンド・ファイルへの参照はすべて現行計画用になる。

【0106】ワークステーションがディスクからブートされた場合、NWICHECKはそのワークステーション用のLCUコマンド・ファイルを実行する。ワークステーションがハードファイルからブートされた場合、NWICHECKは、ハードファイル上にネットワーク導入ユーティリティを導入することになるPREコマンド・ファイルを実行する。システムは、リブートすると、自動的にLCUコマンド・ファイルの実行を開始する。

【0107】LCUコマンド・ファイルは、プログラミングされた導入システムの実行を完了すると、SUCCESSフラグまたはFAILフラグを指定したCheckIt手続きを呼び出す。次に、CheckIt手続きは、現行の計画名とSUCCESSまたはFAILフラグを指定してもう一度NWICHECKを呼び出す。NWICHECKは、計画名を指定して呼び出されると、まず、指定の計画用のPOSTコマンド

・ファイルを実行する。FAILフラグを指定してNWICHECKが呼び出された場合は、処理は停止し、現行計画はそのまま待ち行列に残る。これは、訂正アクションを講じて、ワークステーション上で計画を再始動できるように行われる。成功フラグを指定してNWICHECKが呼び出された場合は、NWICHECKは指定の計画を待ち行列から除去する。待ち行列内に追加の要素がそれ以上ない場合、処理は停止し、追加の要素がまだある場合は、NWICHECKは待ち行列内の第1の要素に関するPREコマンド・ファイルを実行し、システムをリブートする。待ち行列内に追加の要素がそれ以上存在しなくなるか、または計画が失敗に終わるまで、処理は前述のように続行する。

【0108】以下の例について検討する。Wk1とWk2という2つのワークステーションと、Plan1、Plan2、Plan3という3つの計画があるとする。Plan1では、オペレーティング・システムと、トランスポートと、リダイレクタを導入する。Plan2ではアプリケーションを導入し、Plan3ではアプリケーション3を導入する。ワークステーションWk1では、Plan1とPlan2が実行される。ワークステーションWk2では、Plan2とPlan3が実行される。ワークステーションWk1はブート・ディスクから始動され、ワークステーションWk2はハードファイルから始動される。

【0109】ワークステーション1と2のワークステーション待ち行列ファイル：

Wk 1
 Plan1
 Plan2
 Wk 2
 Plan2
 Plan3

【0110】Plan1、Plan2、Plan3に含まれるアプリケーション：

Plan1

オペレーティング・システムを導入する
 トランSPORTを導入する
 リダイレクタを導入する

Plan2

アプリケーション1を導入する

Plan2

アプリケーション2を導入する

【0111】ワークステーションWk 1用の処理：NWICHECKは、wk1の待ち行列を読み取り、Plan1を実行すべきであると判定する。次に、ワークステーションがディスクからブートされたと判定し、Plan1用のwk1 LCU COMMAND FILEを実行する。wk1 LCU COMMAND FILEがオペレーティング・システムと、トランSPORTと、リダイレクタとを導入した後、パラメータとしてPlan1とSUCCESSを指定してNWICHECKが再呼出しされる。NWICHECKは、Plan1用のwk1 POST COMMAND FILEを実行し、次にwk1待ち行列からPlan1を除去する。次にNWICHECKは、Plan2を実行すべきであると判定し、Plan2用のwk1 PRE COMMAND FILEを実行し、システムをリブートする。wk1 LCU COMMAND FILEがアプリケーション1を導入した後、パラメータとしてPlan2とSUCCESSを指定してNWICHECKが再呼出しされる。NWICHECKは、Plan2用のwk1 POST COMMAND FILEを実行し、wk1待ち行列からPlan2を除去する。次にNWICHECKは、wk1待ち行列内に追加の計画がそれ以上存在しないと判定し、処理が停止する。

【0112】ワークステーションWk 2用の処理：NWICHECKは、wk2の待ち行列ファイルを読み取り、Plan2を実行すべきであると判定する。次に、ワークステーションがハードファイルからブートされたと判定し、wk2 PRE COMMAND FILEを実行し、システムをリブートする。wk2 LCU COMMAND FILEがアプリケーション1を導入した後、パラメータとしてPlan2とSUCCESSを指定してNWICHECKが再呼出しされる。NWICHECKは、Plan2用のwk2 POST COMMAND FILEを実行し、次にwk2待ち行列からPlan2を除去する。次にNWICHECKは、Plan3を実行すべきであると判定し、Plan3用のwk2 PRE COMMAND FILEを実行し、システムをリブートする。wk2 LCU COMMAND FILEがアプリケーション2を導入した後、NWICHECKは、Plan3用のwk2 POST COMMAND FILEを実行し、wk2待ち行列からPlan3を除去する。次にNWICHECKは、wk2待ち行列内に追加の計画が

それ以上存在しないと判定し、処理が停止する。

【0113】まとめとして、本発明の構成に関して以下の事項を開示する。

【0114】(1) ネットワークでアプリケーションを導入するための導入計画オブジェクトをコミットする方法において、導入計画オブジェクト内の子オブジェクトを検査し、必要であれば、導入計画オブジェクトに追加の子オブジェクトを追加することにより、導入計画オブジェクトを事前妥当性検査するステップと、導入計画オブジェクトとその子オブジェクト内のデータを検査して、データにエラーがないかどうか確認することにより、導入計画オブジェクトを妥当性検査するステップと、導入計画が正常に妥当性検査された場合に、導入計画オブジェクト内の子オブジェクトを、データ構造に応じてネットワークでアプリケーションを導入するネットワーク導入エンジンに使用可能なデータ構造に変換するステップとを含む方法。

(2) 導入計画オブジェクトが、アプリケーション・プログラムを表す計画中アプリケーション・オブジェクトと、アプリケーション・プログラムを導入すべきワークステーションのグループを表す計画中グループ・オブジェクトとを含むことを特徴とする、上記(1)に記載の方法。

(3) 導入計画が、アプリケーションの導入用のデータを含む応答ファイル・オブジェクトと、特定のワークステーション用の応答ファイル・オブジェクト・データをカスタマイズするためのデータを含むカスタマイズ・ファイル・オブジェクトとをさらに含むことを特徴とする、上記(2)に記載の方法。

(4) 妥当性検査ステップが、導入計画オブジェクト内のデータとして指定されたターゲット・ワークステーション上のファイル・ディレクトリを検査して、そのファイル・ディレクトリが物理的に有効でネットワーク上でアクセス可能であるかどうかを判定することをさらに含み、ターゲット・ワークステーションが、そのアプリケーションを導入すべきワークステーションのグループのうちの1つであることを特徴とする、上記(2)に記載の方法。

(5) 妥当性検査ステップが、ワークステーションのグループ上でのアプリケーションの導入に必要なコード・サーバ・ワークステーション上のオブジェクトの存在を確認することをさらに含み、コード・サーバ・オブジェクトが、導入計画オブジェクトの子オブジェクトではないことを特徴とする、上記(2)に記載の方法。

(6) 妥当性検査ステップが、アプリケーション・オブジェクトが、ネットワーク内の第1のアプリケーションに更新内容を提供するアプリケーションを表していることを判定するステップと、アプリケーション・コンテナ・オブジェクト内の第1のアプリケーションを表す第1のアプリケーション・オブジェクトの存在を確認するス

テップとをさらに含むことを特徴とする、上記(2)に記載の方法。

(7) 妥当性検査ステップが、ターゲット・ワークステーション上のオペレーティング・システムとその訂正サービスのレベルとを確認することをさらに含むことを特徴とする、上記(2)に記載の方法。

(8) 導入計画オブジェクトが、導入すべき複数のアプリケーションと対応する応答ファイルとをそれぞれ表す、複数の計画申請アプリケーション・オブジェクトと複数の対応する応答ファイル・オブジェクトとを含み、変換ステップが、ワークステーションのグループ内の特定のワークステーションに応じて複数の応答ファイル・オブジェクトをカスタマイズするために、カスタマイズ・ファイル・オブジェクトを複数の応答ファイル・オブジェクトに適用することをさらに含むことを特徴とする、上記(3)に記載の方法。

(9) 複数の導入計画オブジェクトについて、事前妥当性検査ステップと、妥当性検査ステップと、変換ステップとを繰り返すステップと、各ワークステーションごとにワークステーション待ち行列に複数の導入計画オブジェクトを待ち行列化するステップとをさらに含むことを特徴とする、上記(1)に記載の方法。

(10) ネットワークでアプリケーションを導入するための導入計画オブジェクトをコミットするシステムにおいて、導入計画オブジェクト内の子オブジェクトを検査し、必要であれば、導入計画オブジェクトに追加の子オブジェクトを追加することにより、導入計画オブジェクトを事前妥当性検査する手段と、導入計画オブジェクトとその子オブジェクト内のデータを検査して、データにエラーがないかどうか確認することにより、導入計画オブジェクトを妥当性検査する手段と、導入計画が正常に妥当性検査された場合に、導入計画オブジェクト内の子オブジェクトを、データ構造に応じてネットワークでアプリケーションを導入するネットワーク導入エンジンに使用可能なデータ構造に変換する手段とを含むシステム。

(11) 導入計画オブジェクトが、アプリケーション・プログラムを表す計画申請アプリケーション・オブジェクトと、アプリケーション・プログラムを導入すべきワークステーションのグループを表す計画申請グループ・オブジェクトとを含むことを特徴とする、上記(10)に記載のシステム。

(12) 導入計画が、アプリケーションの導入用のデータを含む応答ファイル・オブジェクトと、特定のワークステーション用の応答ファイル・オブジェクト・データをカスタマイズするためのデータを含むカスタマイズ・ファイル・オブジェクトとをさらに含むことを特徴とする、上記(11)に記載のシステム。

(13) 妥当性検査手段が、導入計画オブジェクト内のデータとして指定されたターゲット・ワークステーション

上のファイル・ディレクトリを検査して、そのファイル・ディレクトリが物理的に有効でネットワーク上でアクセス可能であるかどうかを判定する手段をさらに含み、ターゲット・ワークステーションが、そのアプリケーションを導入すべきワークステーションのグループのうちの1つであることを特徴とする、上記(11)に記載のシステム。

(14) 妥当性検査手段が、ワークステーションのグループ上でのアプリケーションの導入に必要なコード・サーバ・ワークステーション上のオブジェクトの存在を確認する手段をさらに含み、コード・サーバ・オブジェクトが、導入計画オブジェクトの子オブジェクトではないことを特徴とする、上記(11)に記載のシステム。

(15) 妥当性検査手段が、アプリケーション・オブジェクトが、ネットワーク内の第1のアプリケーションに更新内容を提供するアプリケーションを表していることを判定する手段と、アプリケーション・コンテナ・オブジェクト内の第1のアプリケーションを表す第1のアプリケーション・オブジェクトの存在を確認する手段とをさらに含むことを特徴とする、上記(11)に記載のシステム。

(16) 導入計画オブジェクトが、導入すべき複数のアプリケーションと対応する応答ファイルとをそれぞれ表す、複数の計画申請アプリケーション・オブジェクトと複数の対応する応答ファイル・オブジェクトとを含み、変換ステップが、ワークステーションのグループ内の特定のワークステーションに応じて複数の応答ファイル・オブジェクトをカスタマイズするために、カスタマイズ・ファイル・オブジェクトを複数の応答ファイル・オブジェクトに適用することをさらに含むことを特徴とする、上記(12)に記載のシステム。

(17) 複数の導入計画オブジェクトについて、事前妥当性検査ステップと、妥当性検査ステップと、変換ステップとを繰り返す手段と、各ワークステーションごとにワークステーション待ち行列に複数の導入計画オブジェクトを待ち行列化する手段とをさらに含むことを特徴とする、上記(10)に記載のシステム。

(18) ネットワークでアプリケーションを導入するための導入計画オブジェクトをコミットするためにコンピュータで読取り可能な命令を格納するコンピュータ・メモリにおいて、導入計画オブジェクト内の子オブジェクトを検査し、必要であれば、導入計画オブジェクトに追加の子オブジェクトを追加することにより、導入計画オブジェクトを事前妥当性検査する手段と、導入計画オブジェクトとその子オブジェクト内のデータを検査して、データにエラーがないかどうか確認することにより、導入計画オブジェクトを妥当性検査する手段と、導入計画が正常に妥当性検査された場合に、導入計画オブジェクト内の子オブジェクトを、データ構造に応じてネットワークでアプリケーションを導入するネットワーク導入エ

ンジンに使用可能なデータ構造に変換する手段とを含み、コンピュータ・メモリがコンピュータ・システムに結合され、コンピュータ・システムによってアクセスされたときに、その手段が活動化されることを特徴とするコンピュータ・メモリ。

(19) 導入計画オブジェクトが、アプリケーション・プログラムを表す計画中アプリケーション・オブジェクトと、アプリケーション・プログラムを導入すべきワークステーションのグループを表す計画中グループ・オブジェクトとを含むことを特徴とする、上記(18)に記載のコンピュータ・メモリ。

(20) 導入計画が、アプリケーションの導入用のデータを含む応答ファイル・オブジェクトと、特定のワークステーション用の応答ファイル・オブジェクト・データをカスタマイズするためのデータを含むカスタマイズ・ファイル・オブジェクトとをさらに含むことを特徴とする、上記(19)に記載のコンピュータ・メモリ。

(21) 妥当性検査手段が、導入計画オブジェクト内のデータとして指定されたターゲット・ワークステーション上のファイル・ディレクトリを検査して、そのファイル・ディレクトリが物理的に有効でネットワーク上でアクセス可能であるかどうかを判定する手段とをさらに含み、ターゲット・ワークステーションが、そのアプリケーションを導入すべきワークステーションのグループのうちの一つであることを特徴とする、上記(19)に記載のコンピュータ・メモリ。

(22) 妥当性検査手段が、ワークステーションのグループ上でのアプリケーションの導入に必要なコード・サーバ・ワークステーション上のオブジェクトの存在を確認する手段とをさらに含み、コード・サーバ・オブジェクトが、導入計画オブジェクトの子オブジェクトではないことを特徴とする、上記(19)に記載のコンピュータ・メモリ。

(23) 妥当性検査手段が、アプリケーション・オブジェクトが、ネットワーク内の第1のアプリケーションに更新内容を提供するアプリケーションを表していることを判定する手段と、アプリケーション・コンテナ・オブジェクト内の第1のアプリケーションを表す第1のアプリケーション・オブジェクトの存在を確認する手段とをさらに含むことを特徴とする、上記(19)に記載のコンピュータ・メモリ。

(24) 導入計画オブジェクトが、導入すべき複数のアプリケーションと対応する応答ファイルとをそれぞれ表す、複数の計画中アプリケーション・オブジェクトと複数の対応する応答ファイル・オブジェクトとを含み、変換ステップが、ワークステーションのグループ内の特定のワークステーションに応じて複数の応答ファイル・オブジェクトをカスタマイズするために、カスタマイズ・ファイル・オブジェクトを複数の応答ファイル・オブジェクトに適用することをさらに含むことを特徴とする、

上記(20)に記載のコンピュータ・メモリ。

(25) 複数の導入計画オブジェクトについて、事前妥当性検査ステップと、妥当性検査ステップと、変換ステップとを繰り返す手段と、各ワークステーションごとにワークステーション待ち行列に複数の導入計画オブジェクトを待ち行列化する手段とをさらに含むことを特徴とする、上記(18)に記載のコンピュータ・メモリ。

【図面の簡単な説明】

【図1】本発明の教示により構成したコンピュータ・システムを示す図である。

【図2】導入計画オブジェクトの階層表現を示す図である。

【図3】導入計画オブジェクト内のオブジェクトに関連するオブジェクトを示す図である。

【図4】導入計画オブジェクト内のオブジェクトに関連するオブジェクトを示す図である。

【図5】導入計画オブジェクト内のオブジェクトに関連するオブジェクトを示す図である。

【図6】導入計画オブジェクト内のオブジェクトに関連するオブジェクトを示す図である。

【図7】コミット・プロセスの流れ図である。

【図8】事前妥当性検査プロセスの流れ図である。

【図9】導入計画オブジェクトに関する妥当性検査プロセスの流れ図である。

【図10】導入計画オブジェクト内の応答ファイルにカスタマイズ・ファイル・オブジェクトを適用するプロセスを示す図である。

【図11】計画中アプリケーション・オブジェクトに関する妥当性検査プロセスを示す図である。

【図12】計画中アプリケーション・オブジェクトに保守システムが必要かどうかを判定するためのプロセスの流れ図である。

【図13】計画に含まれないが計画内のオブジェクトに関連するアプリケーション・オブジェクトに関する妥当性検査プロセスの流れ図である。

【図14】計画中ワークステーション・オブジェクトに関する妥当性検査プロセスを示す図である。

【図15】計画中ワークステーション・アプリケーション・オブジェクトに関する妥当性検査プロセスを示す図である。

【図16】計画中カテゴリ・オブジェクトに関する妥当性検査プロセスの流れ図である。

【図17】応答ファイルとカスタマイズ・ファイル・オブジェクトの妥当性検査を示す図である。

【図18】計画中グループ・オブジェクトの妥当性検査の流れ図である。

【図19】コード・サーバ・オブジェクトに関する妥当性検査プロセスを示す図である。

【図20】コード・サーバ・オブジェクト用の計画中アプリケーション・オブジェクトについてアプリケーション

ン・イメージが存在するかどうかを検査するためのプロセスを示す図である。

- 【図21】計画出力ファイルを示す図である。
- 【図22】計画状況ファイルを示す図である。
- 【図23】ワークステーション待ち行列ファイルを示す図である。
- 【図24】PREコマンド・ファイルを示す図である。
- 【図25】POSTコマンド・ファイルを示す図である。
- 【図26】LCUコマンド・ファイルを示す図である。
- 【図27】導入コマンド・スクリプトを生成するためのプロセスの流れ図である。
- 【図28】LCUコマンド作成プロセスの流れ図である。
- 【図29】可変セクションを書き込むためのプロセスを示す図である。
- 【図30】製品データ・セクションを書き込むためのプロセスを示す図である。
- 【図31】書き込み検査手続きを示す図である。
- 【図32】導入セクションを書き込むための流れ図である。
- 【図33】アプリケーション・ルーチンを書き込むための流れ図である。
- 【図34】導入実行書き込みルーチンの流れ図である。
- 【図35】最終アプリケーション開始ルーチンの流れ図である。
- 【図36】終了セクション開始ルーチンの流れ図である。

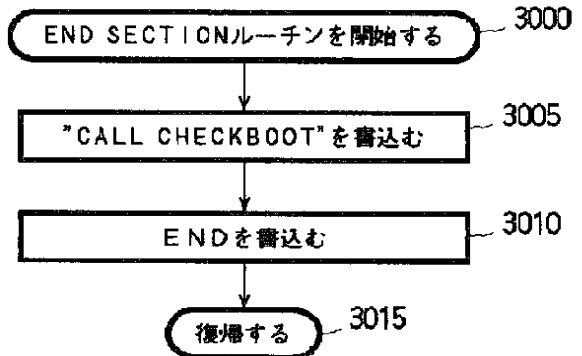
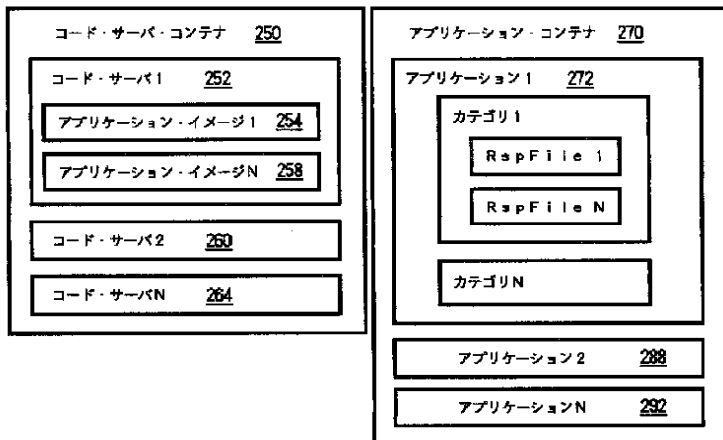
る。

- 【図37】保守セクション・プロセスを示す図である。
- 【図38】ブート・ディスク・プロセスを示す図である。
- 【図39】ハード・ディスク・プロセスを示す図である。
- 【符号の説明】
- 600 導入計画用のVALIDATEを開始する
- 605 計画が少なくとも1つのAppIPを含むかどうか
- 610 I = 1
- 615 I <= AppIPの数であるかどうか
- 620 VALIDATEメソッドをAppIP, Iに送る
- 625 I = I + 1
- 630 VALERROR=VALERROR+1
- 635 計画がGroupIPを含むかどうか
- 640 VALIDATEメソッドをGroupIPを送る
- 645 VALERROR=VALERROR+1
- 650 計画用のローカル・ディレクトリが存在するかどうか
- 655 VALERROR=VALERROR+1
- 660 I = 1
- 665 I <= コード・サーバ・オブジェクトの数であるかどうか
- 670 VALIDATEメソッドをCODESERVER, Iに送る
- 675 I = I + 1
- 680 VALIDATEメソッドから復帰する

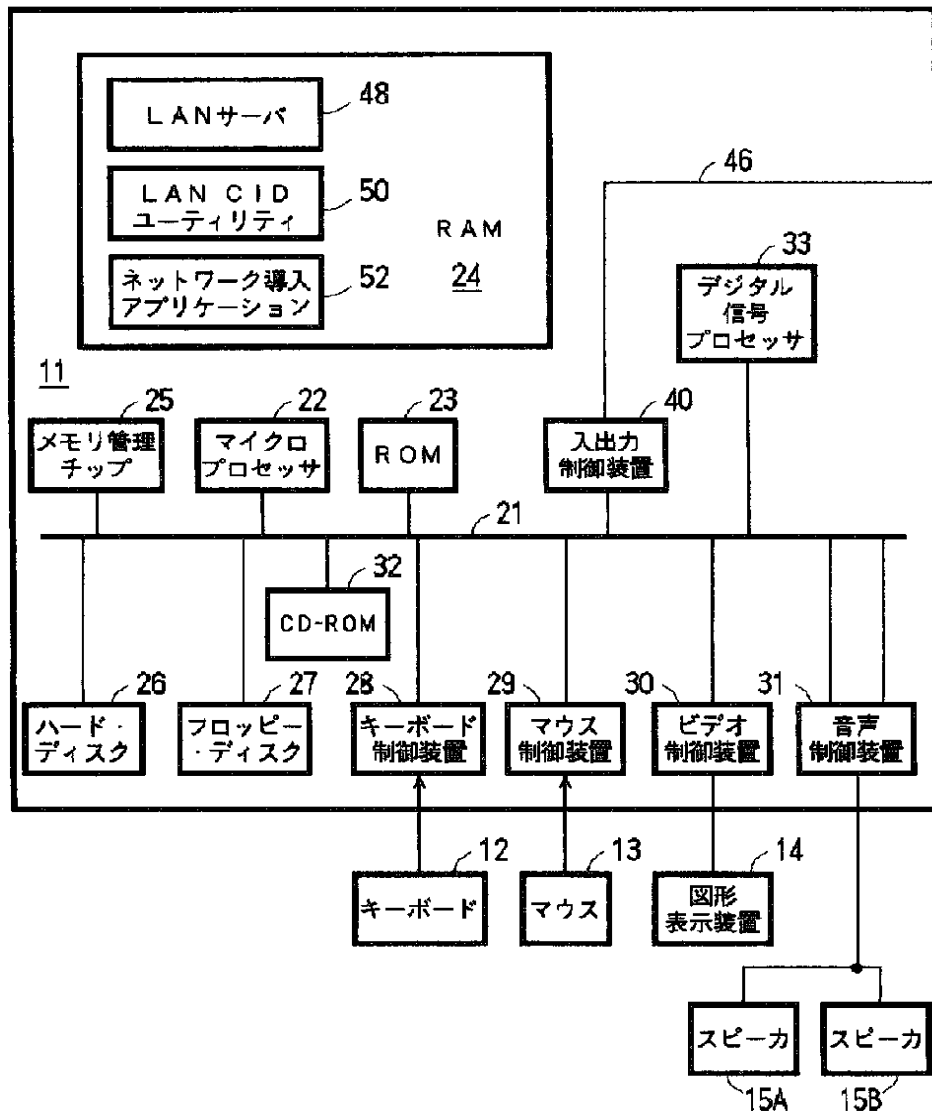
【図3】

【図4】

【図36】



【図1】



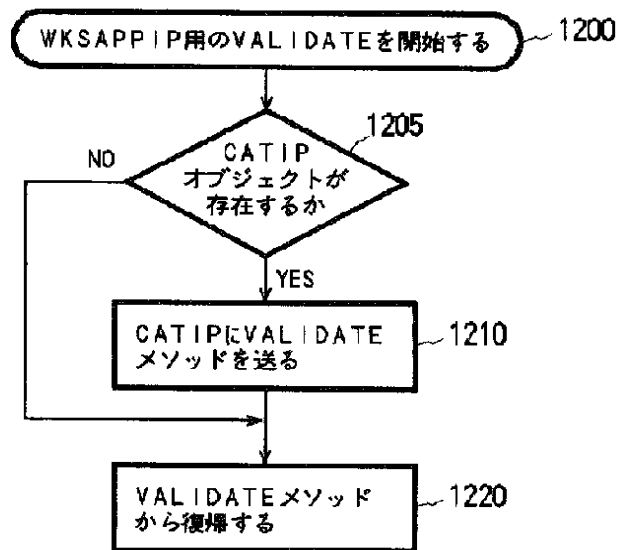
【図2】

計 画	200
APPIP 1	202
CUSTFILE 1	204
CATIP 1	206
RSPFILE 1	208
APPIP 2	210
⋮	212
APPIP N	214
CATIP N	216
RSPFILE N	218
GROUPIP	220
WKSIP 1	222
WKSAPPIP 1	224
CATIP 1	226
RSPFILE 1	228
WKSAPPIP 2	230
⋮	232
WKSAPPIP N	234
CATIP N	236
RSPFILE N	238
WKSIP 2	240
⋮	242
WKSIP M	244

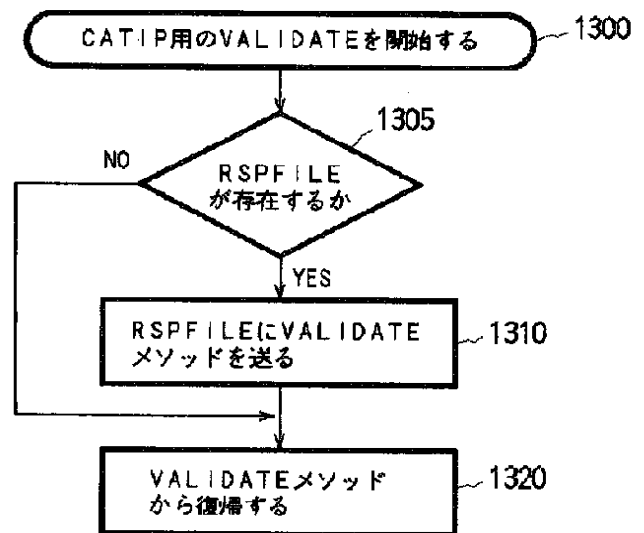
【図23】

ワークステーション待ち行列ファイル	2060
計画名 1	2062
...	2064
計画名 N	2066

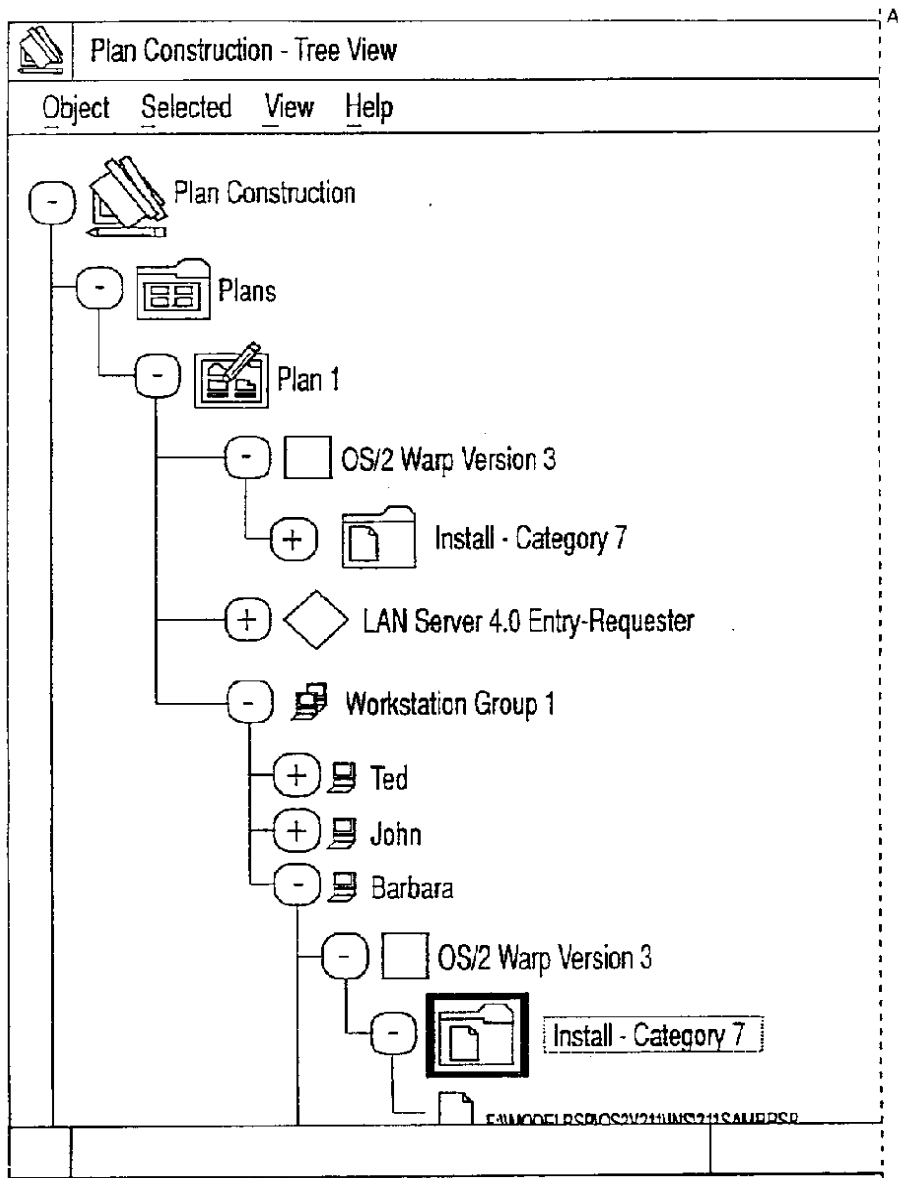
【図15】



【図16】

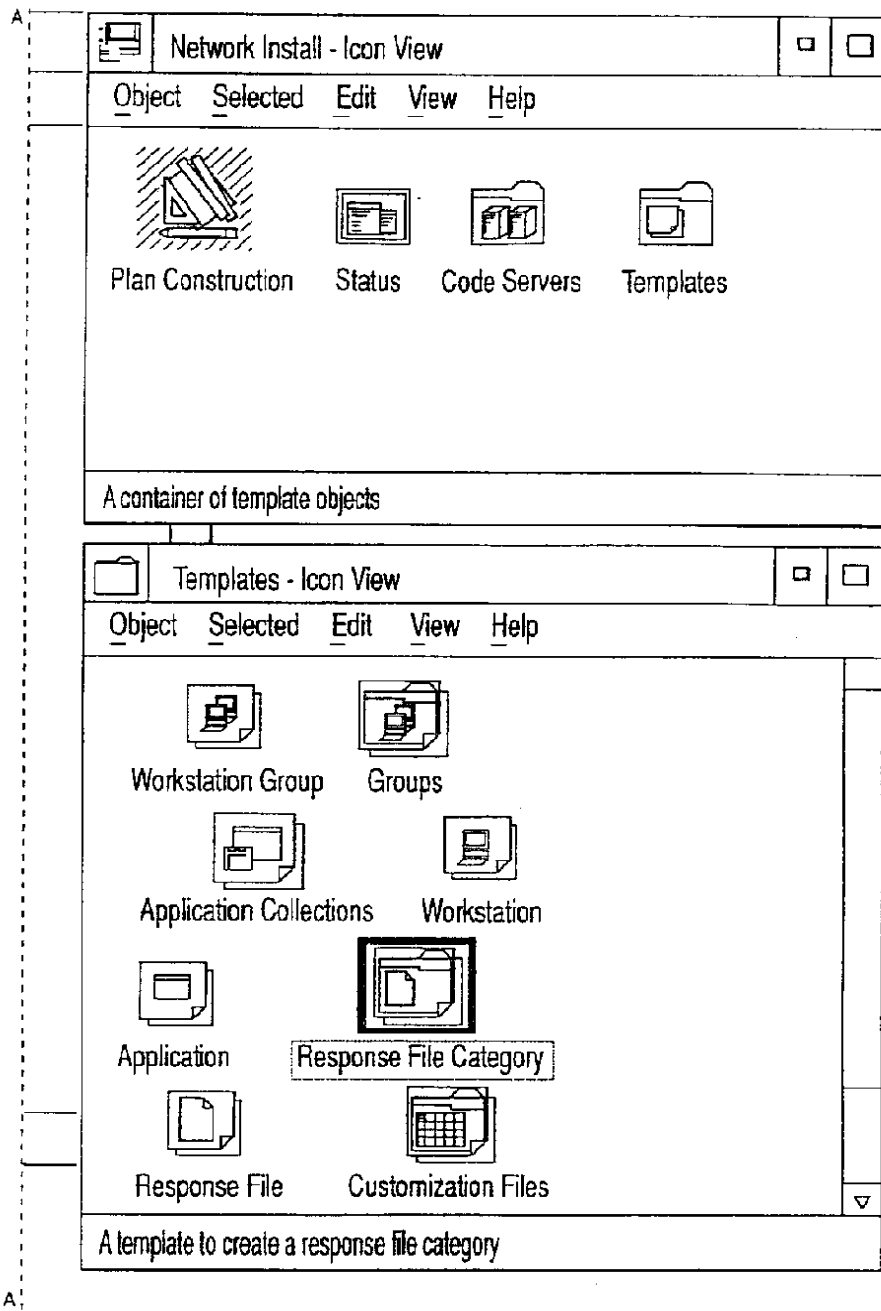


【図5】

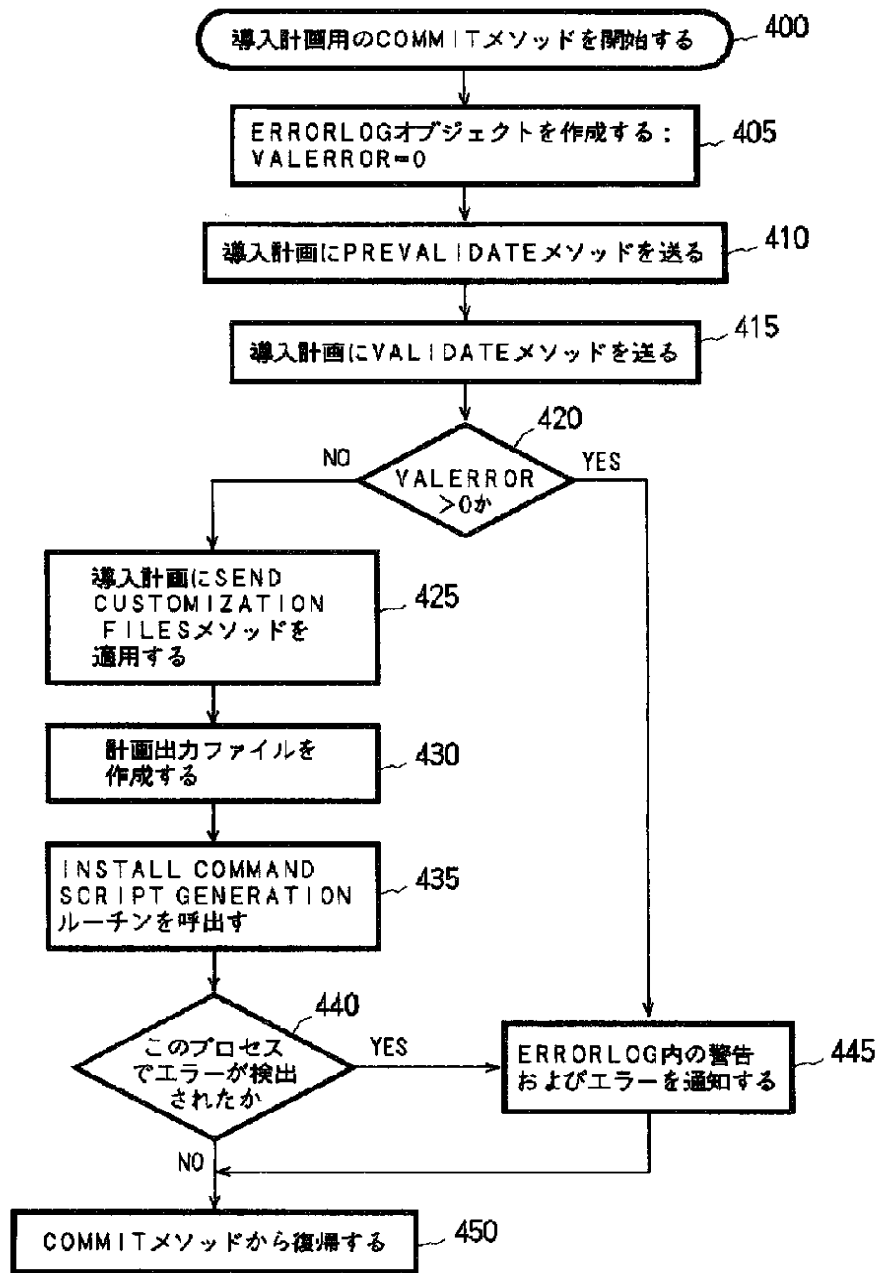


計画中の応答ファイルのコンテナ

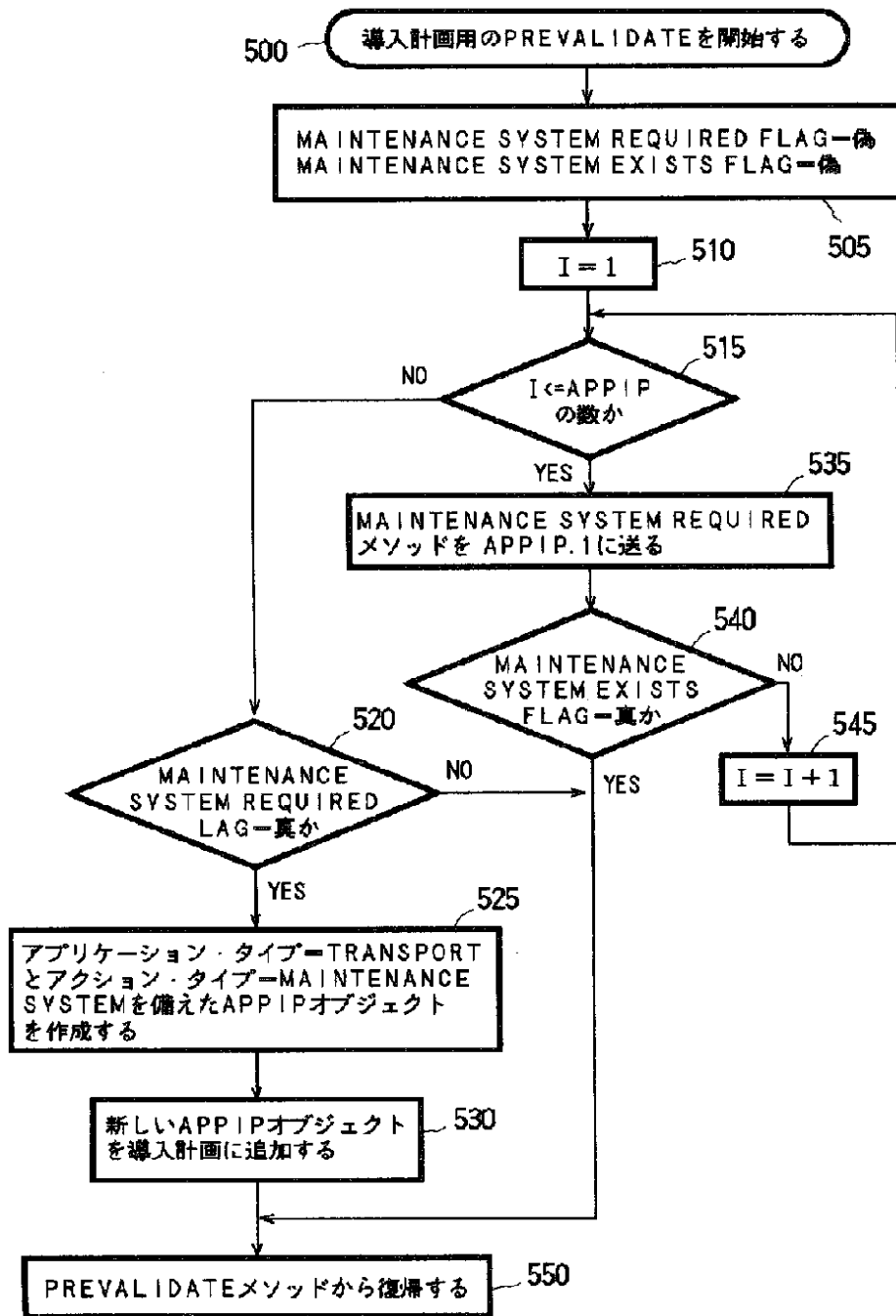
【図6】



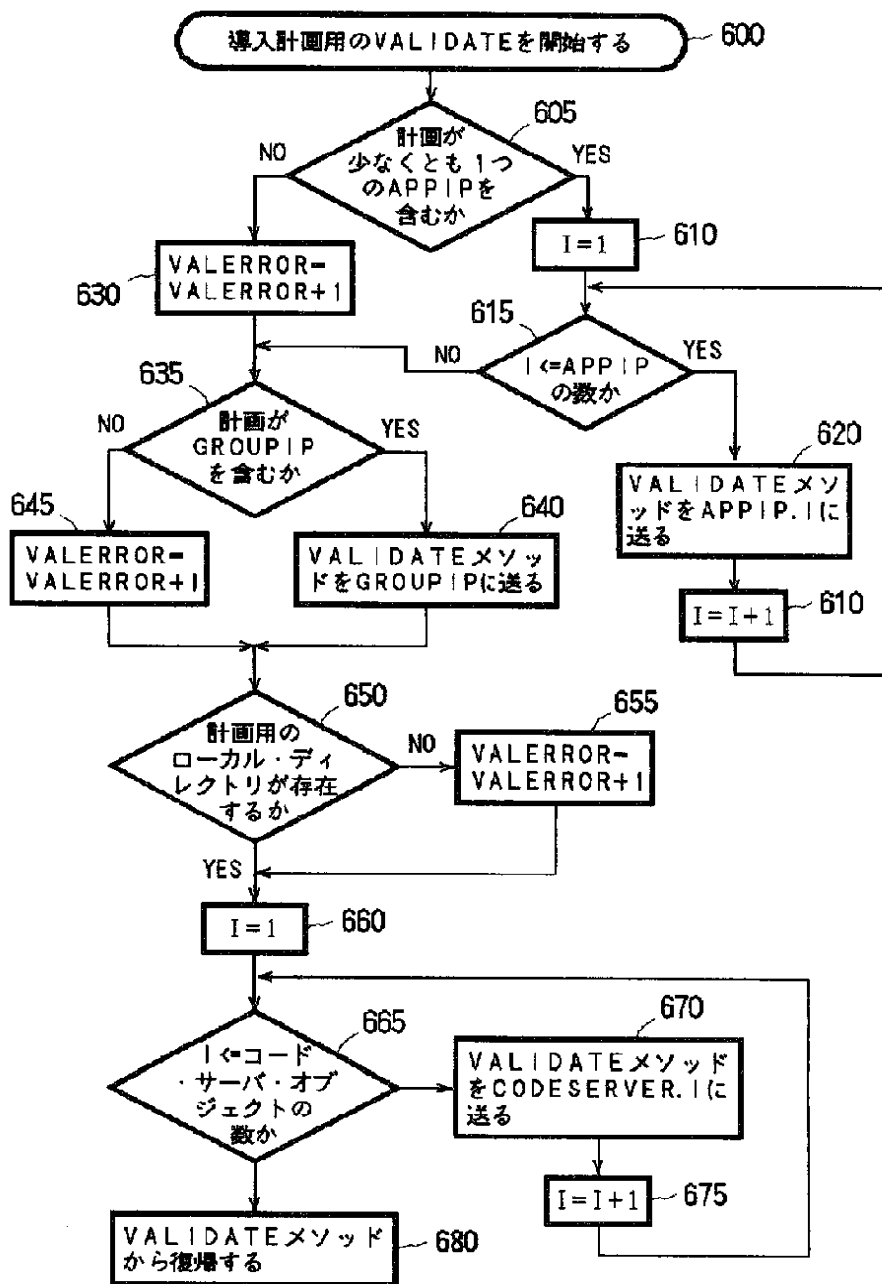
【図7】



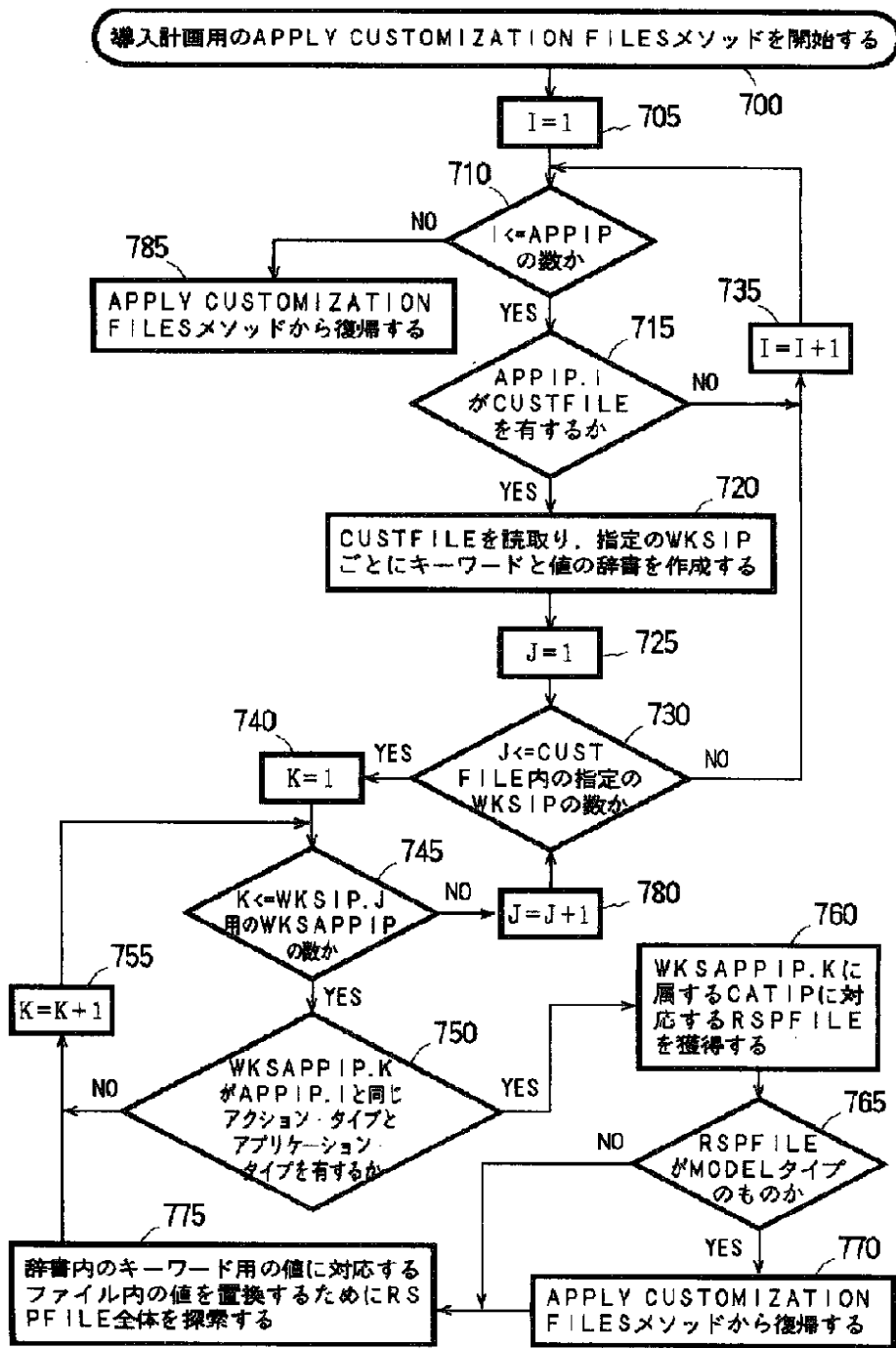
【図8】



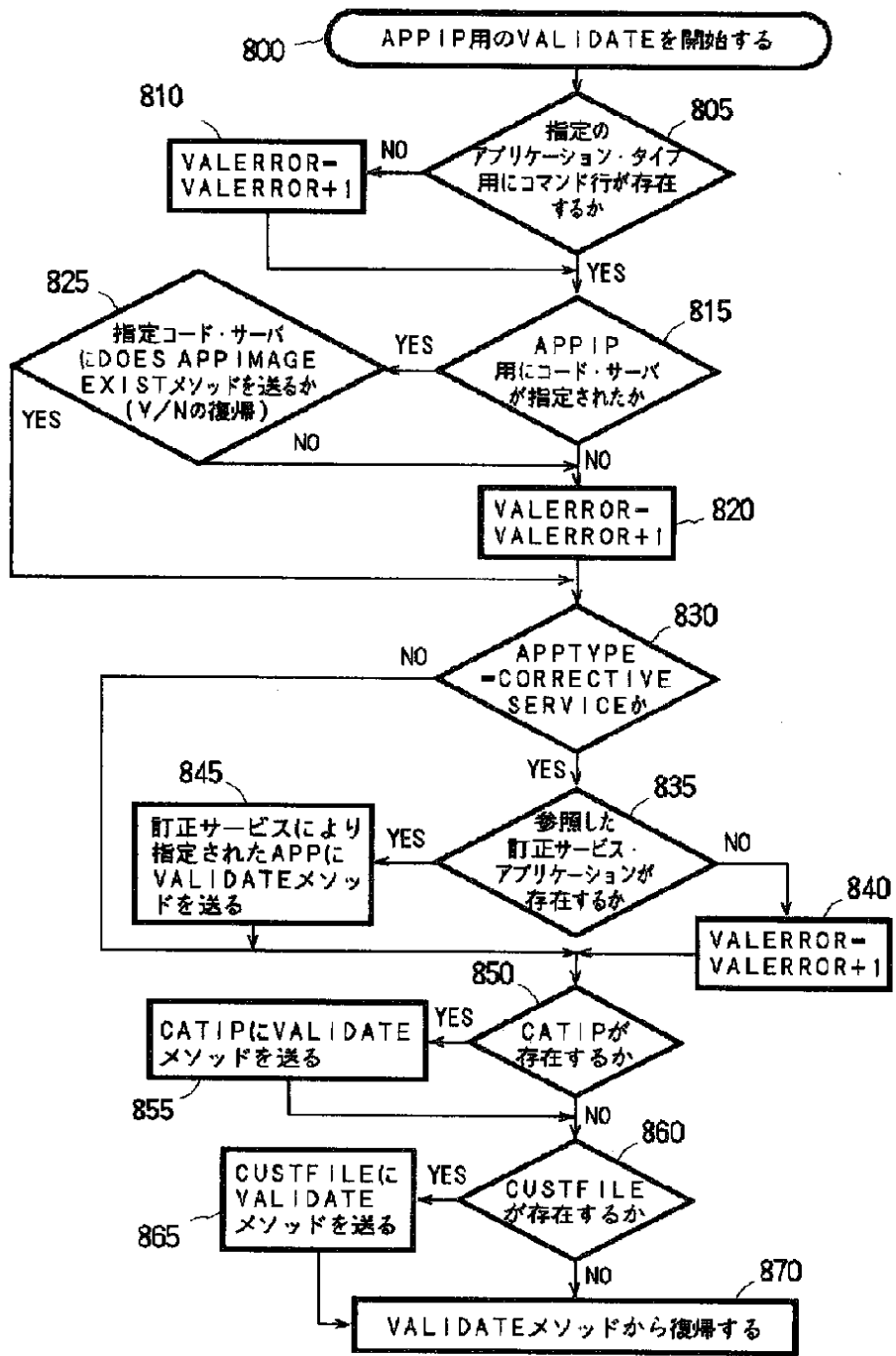
【図9】



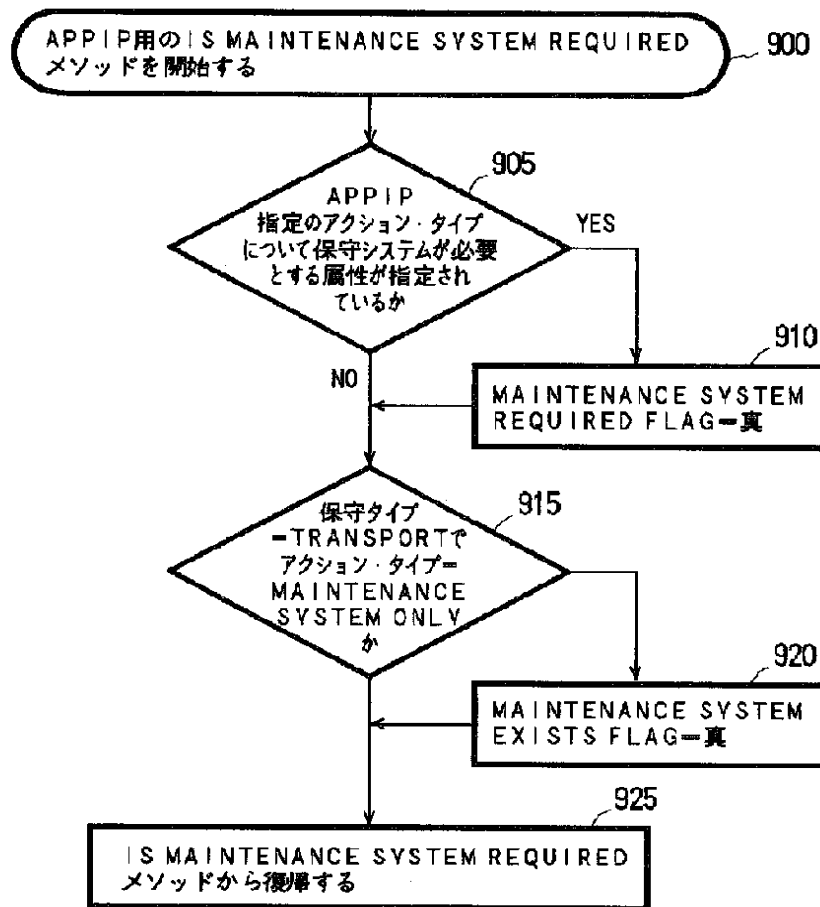
【図10】



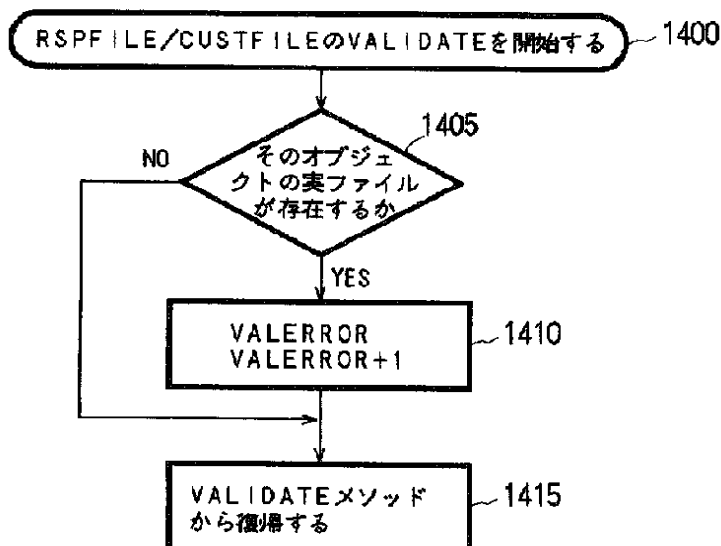
【図11】



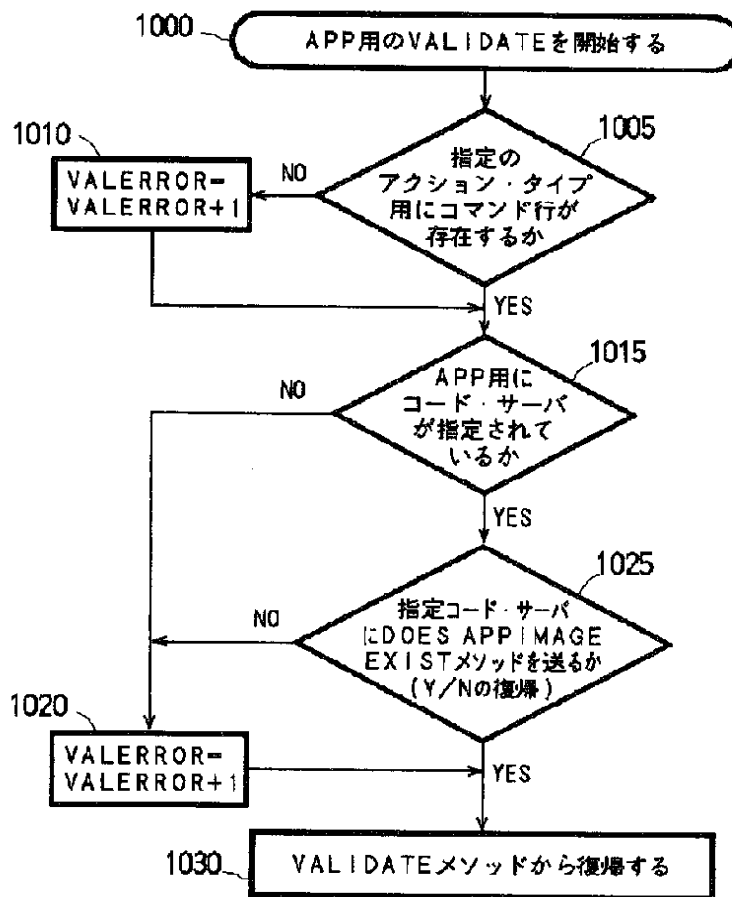
【図12】



【図17】



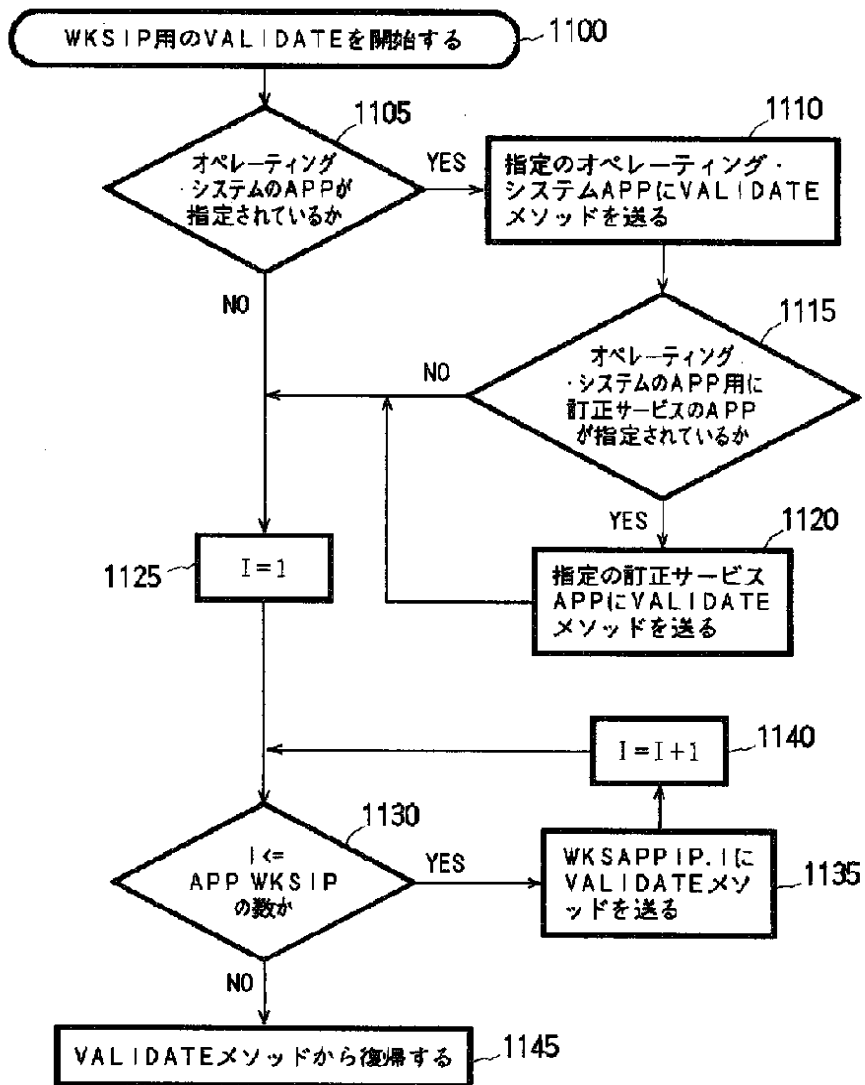
【図13】



【図22】

計画状況ファイル	2030
計画セクション	2032
ワークステーション・セクション1	2034
ワークステーション・ログ1	2036
...	2038
ワークステーション・ログ1	2040
ワークステーション状況ファイル	2042
...	2044
ワークステーション・セクション1	2046

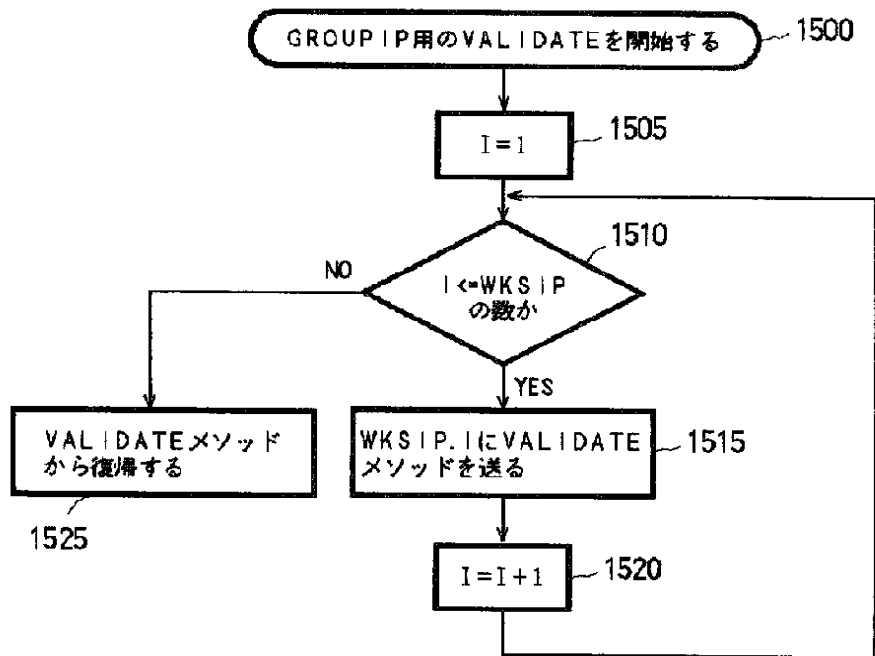
【図14】



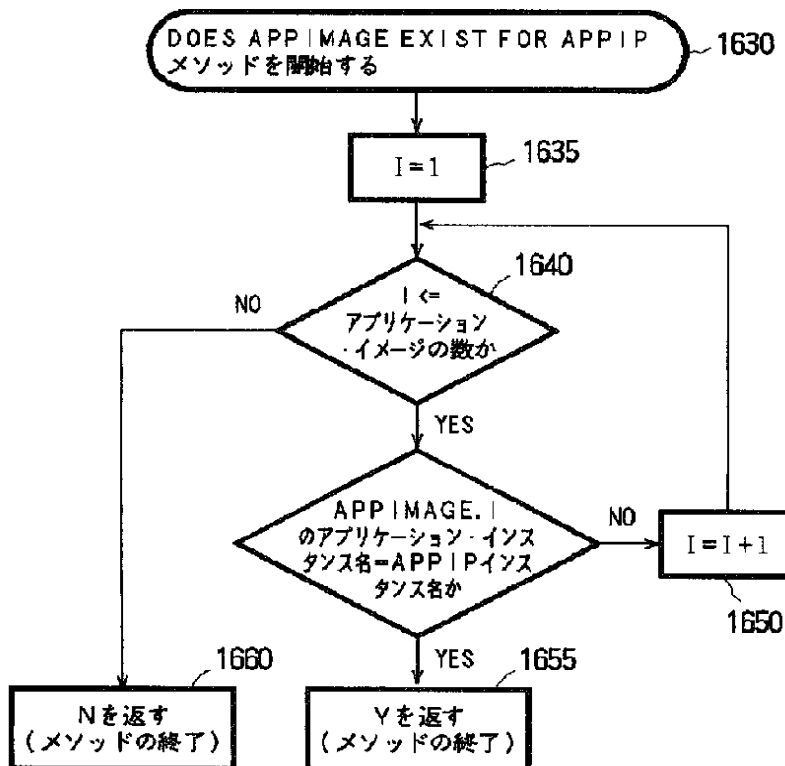
【図24】

PREコマンド・ファイル	2070
接続導入1	2072
...	2074
接続導入N	2076
ネットワーク導入使用可能性	2078

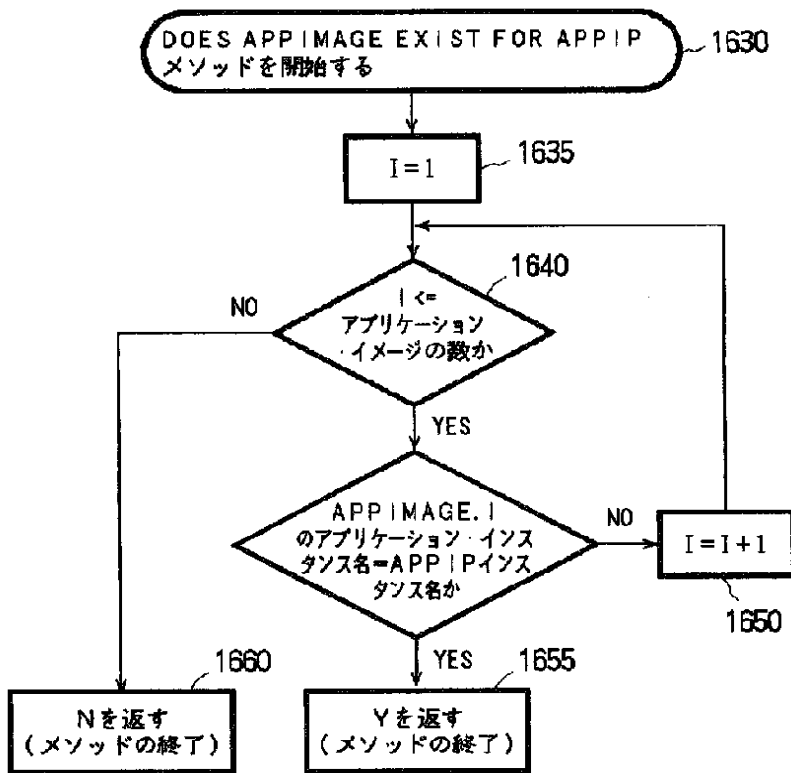
【図18】



【図19】



【図20】



【図21】

POFファイル	2000
計画セクション	2002
シーケンス	2004
APPセクション1	2006
...	2008
APPセクションN	2010
ワークステーション・セクション1	2012
...	2014
ワークステーション・セクションN	2016
サーバ・セクション1	2018
...	2020
サーバ・セクションN	2022

【図35】



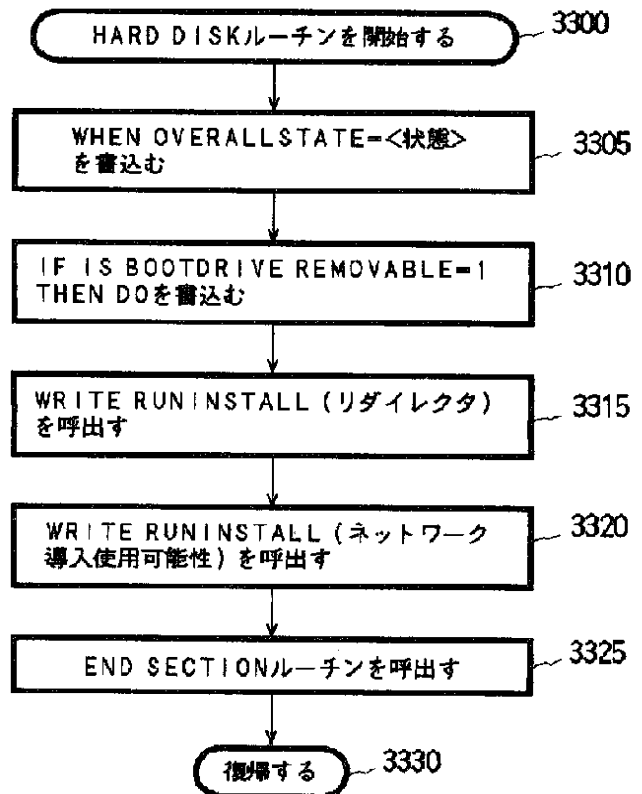
【図25】

POSTコマンド・ファイル	2080
接続除去1	2082
...	2084
接続除去1	2086
ネットワーク導入使用可能除去	2088

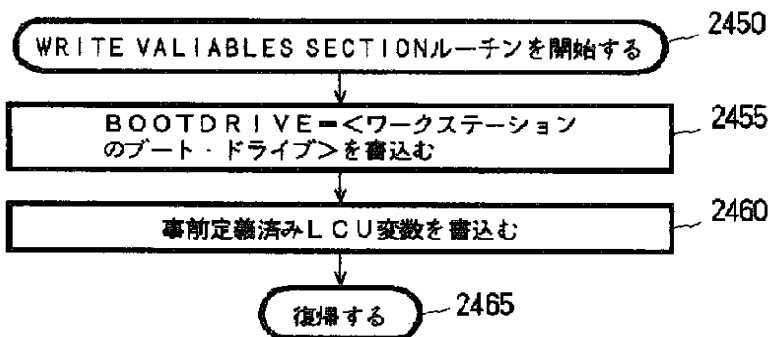
【図26】

LCUコマンド・ファイル	2090
セットアップ・セクション	2092
接続セクション	2094
変数セクション	2096
製品データ・セクション	2098
導入セクション	2100
CheckIT手続き	2102
基本LCU手続きセクション	2104

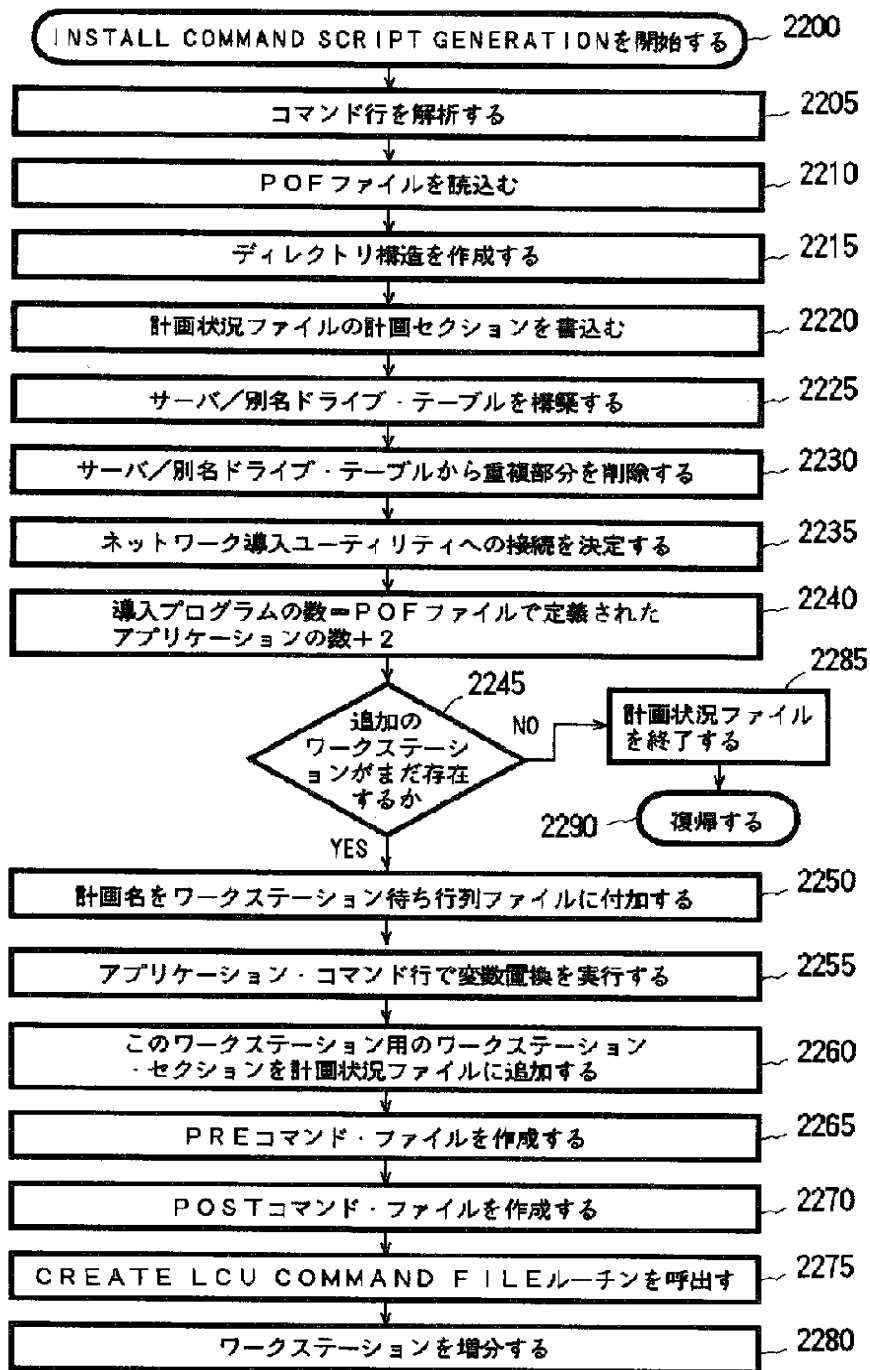
【図39】



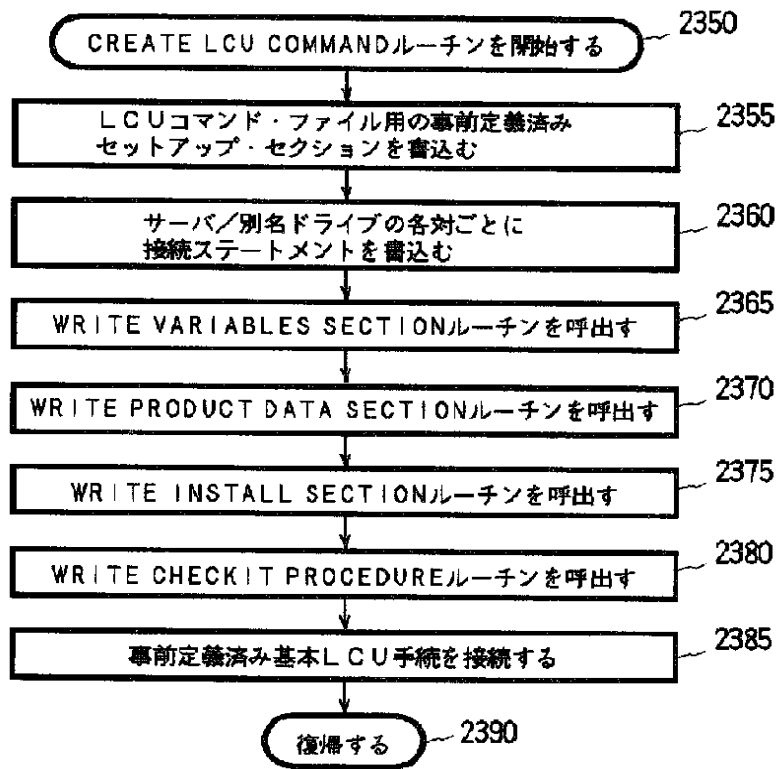
【図29】



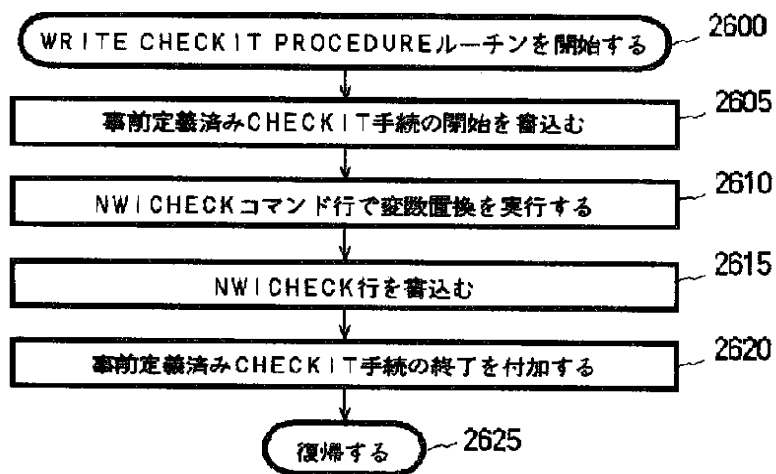
【図27】



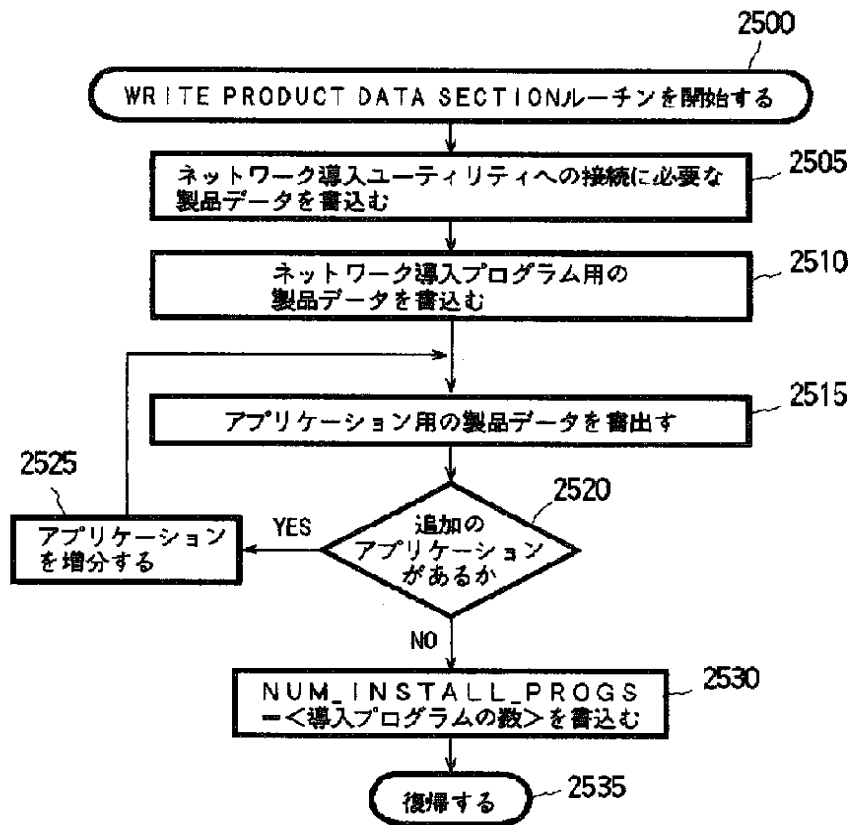
【図28】



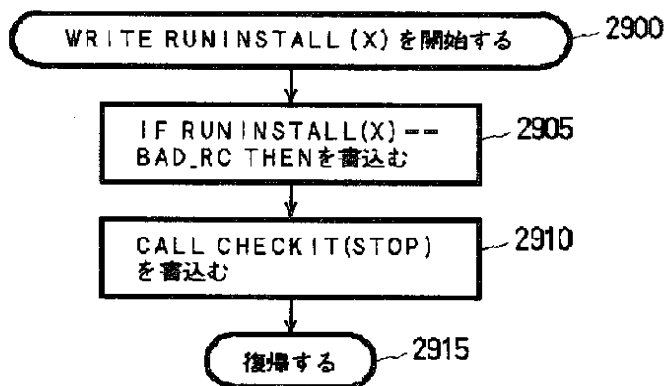
【図31】



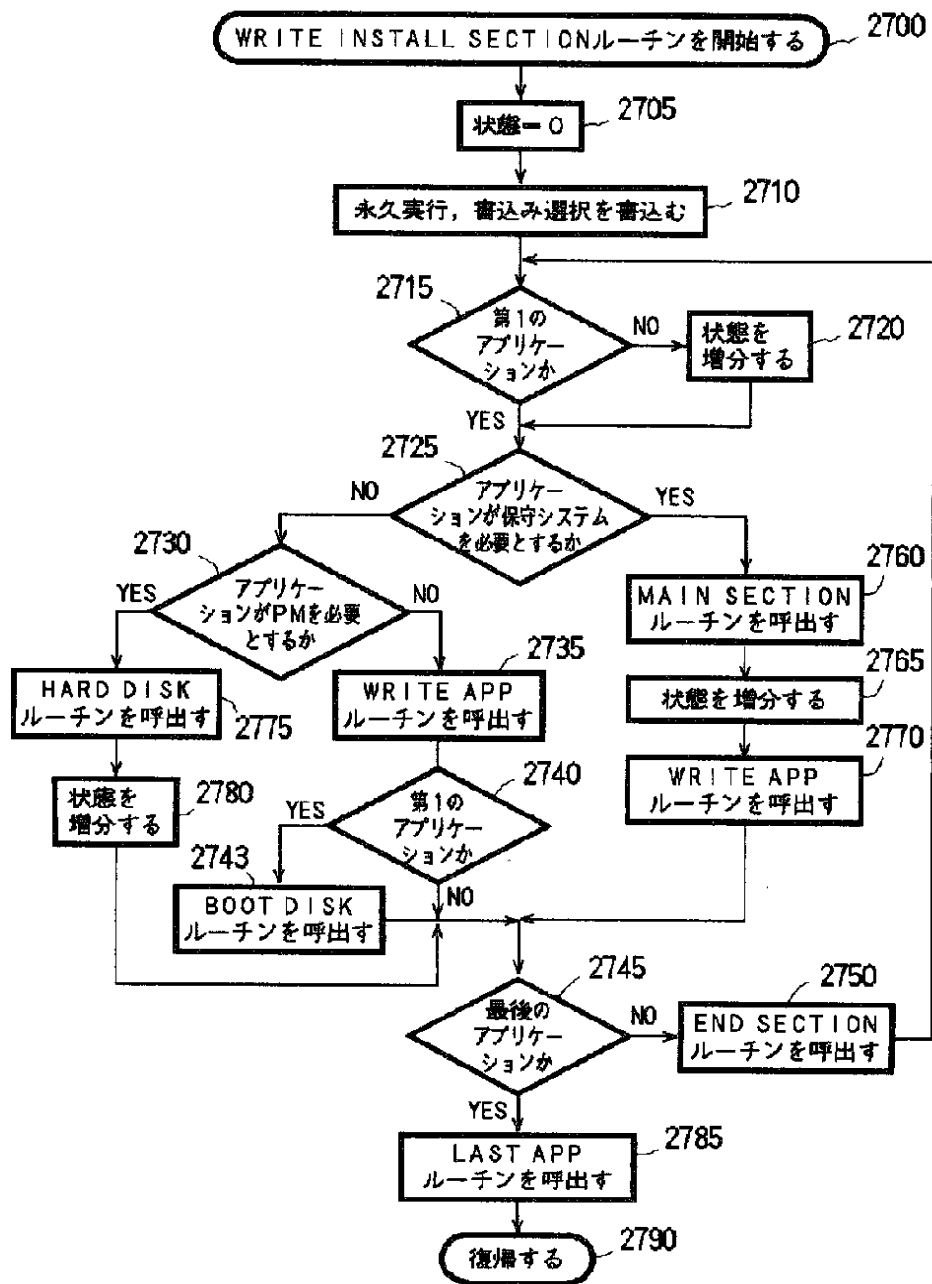
【図30】



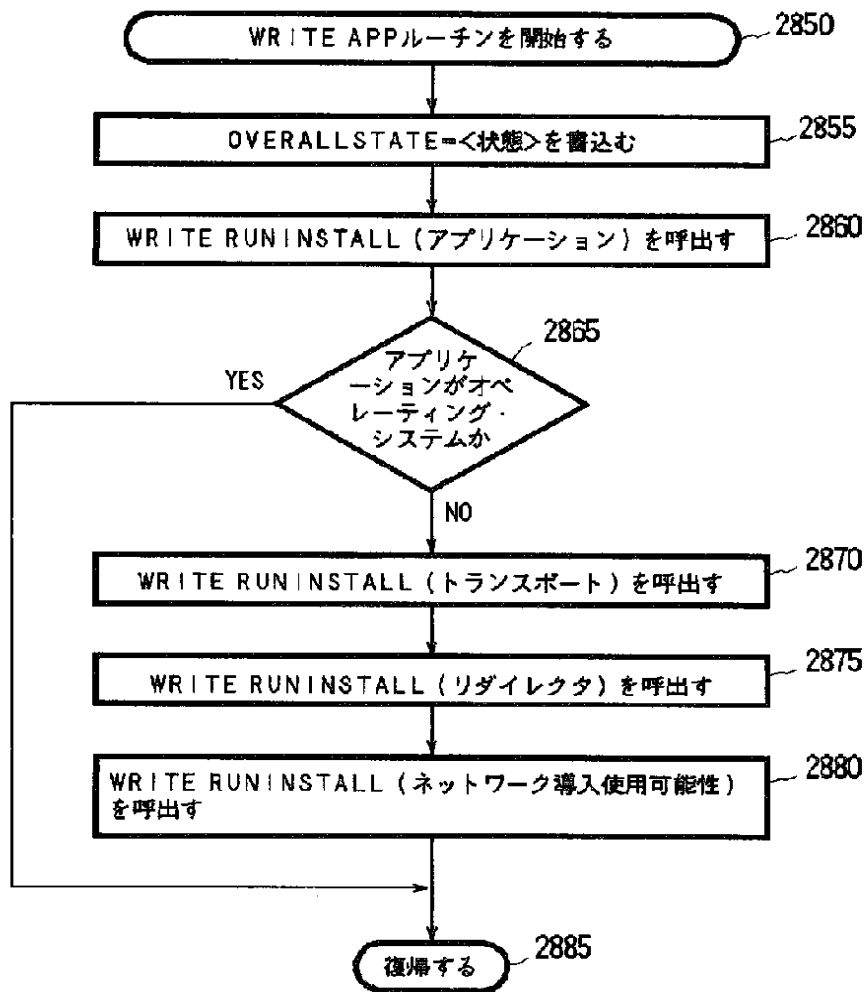
【図34】



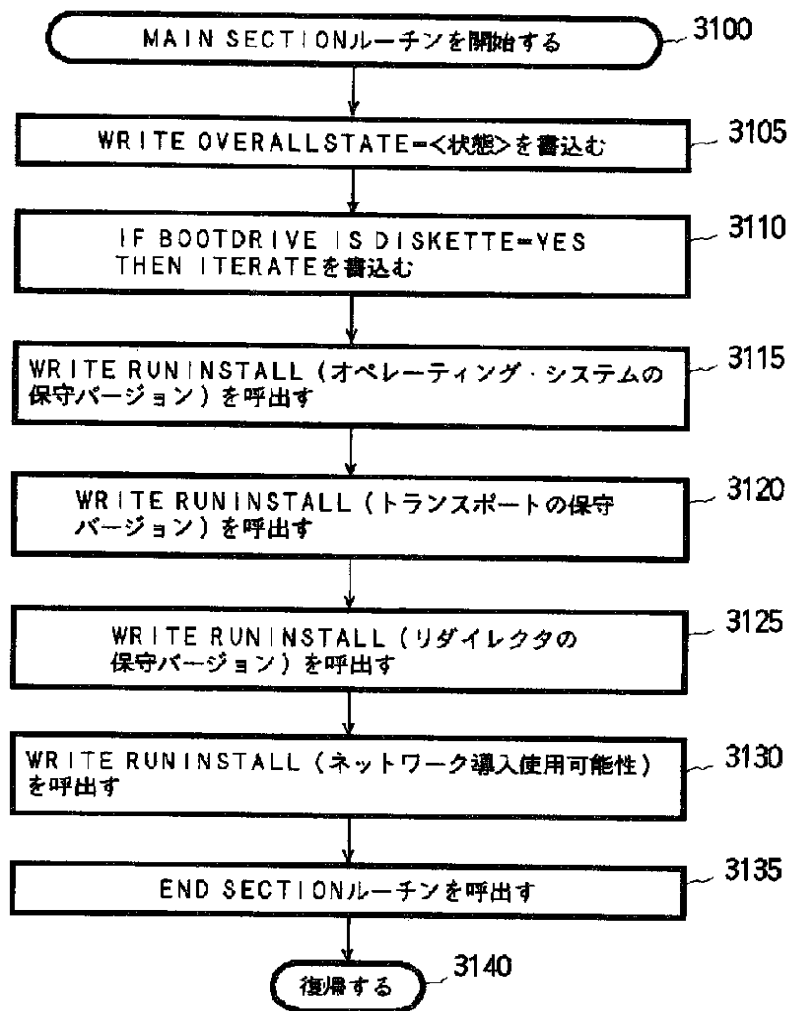
【図32】



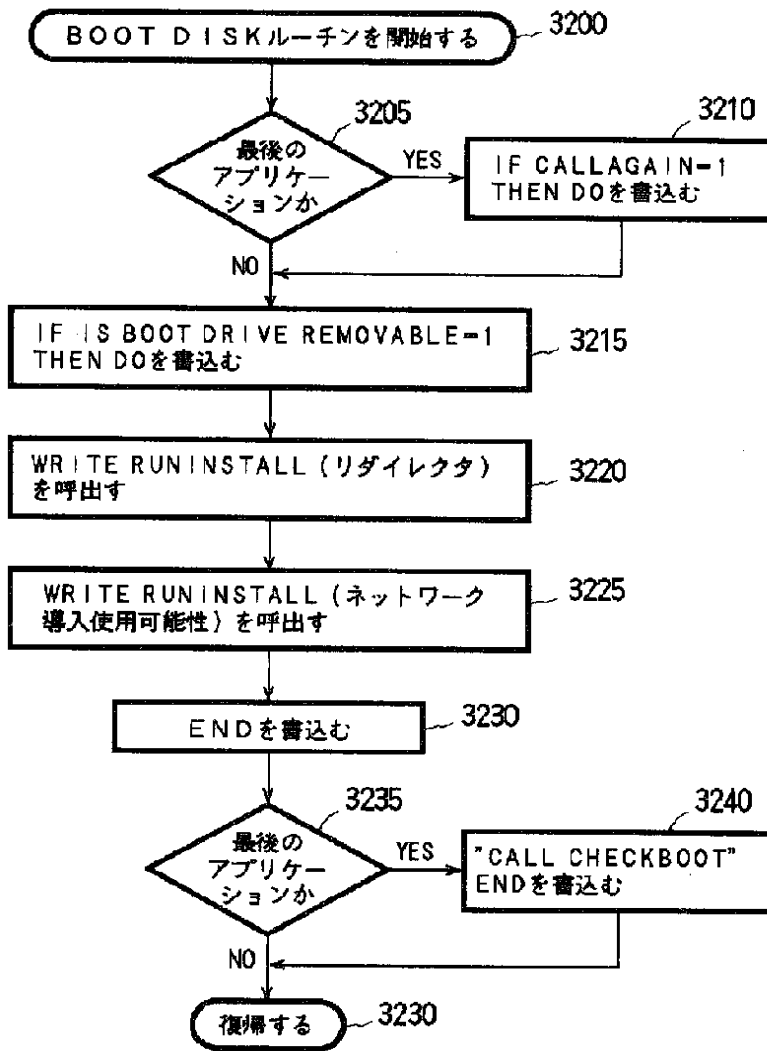
【図33】



【図37】



【図38】



フロントページの続き

(72)発明者 バーバラ・ジャン・ジェンセン
アメリカ合衆国78727-6075 テキサス州
オースチン クレーグ・ドライブ 4902

(72)発明者 セオドア・ジャック・ロンドン・シュレー
ダー
アメリカ合衆国78613 テキサス州シーダ
ー・パーク シェイディ・ブルック・レー
ン 1704